IET

Control Engineering Series 52

Adaptive Prediction and Predictive Control

P.P. Kanjilal

IET CONTROL ENGINEERING SERIES 52

Series Editors: Professor D.P. Atherton
Professor G.I. Irwin

Adaptive Prediction and Predictive Control

Other volumes in this series:

Volume 2 **Elevator traffic analysis, design and control, 2nd edition** G.C. Barney and S.M. dos Santos

Volume 8 **A history of control engineering, 1800–1930** S. Bennett

Volume 14 **Optimal relay and saturating control system synthesis** E.P. Ryan

Volume 18 **Applied control theory, 2nd edition** J.R. Leigh

Volume 20 **Design of modern control systems** D.J. Bell, P.A. Cook and N. Munro (Editors)

Volume 28 **Robots and automated manufacture** J. Billingsley (Editor)

Volume 32 **Multivariable control for industrial applications** J. O'Reilly (Editor)

Volume 33 **Temperature measurement and control** J.R. Leigh

Volume 34 **Singular perturbation methodology in control systems** D.S. Naidu

Volume 35 **Implementation of self-tuning controllers** K. Warwick (Editor)

Volume 37 **Industrial digital control systems, 2nd edition** K. Warwick and D. Rees (Editors)

Volume 39 **Continuous time controller design** R. Balasubramanian

Volume 40 **Deterministic control of uncertain systems** A.S.I. Zinober (Editor)

Volume 41 **Computer control of real-time processes** S. Bennett and G.S. Virk (Editors)

Volume 42 **Digital signal processing: principles, devices and applications** N.B. Jones and J.D.McK. Watson (Editors)

Volume 44 **Knowledge-based systems for industrial control** J. McGhee, M.J. Grimble and A. Mowforth (Editors)

Volume 47 **A history of control engineering, 1930–1956** S. Bennett

Volume 49 **Polynomial methods in optimal control and filtering** K.J. Hunt (Editor)

Volume 50 **Programming industrial control systems using IEC 1131-3** R.W. Lewis

Volume 51 **Advanced robotics and intelligent machines** J.O. Gray and D.G. Caldwell (Editors)

Volume 52 **Adaptive prediction and predictive control** P.P. Kanjilal

Volume 53 **Neural network applications in control** G.W. Irwin, K. Warwick and K.J. Hunt (Editors)

Volume 54 **Control engineering solutions: a practical approach** P. Albertos, R. Strietzel and N. Mort (Editors)

Volume 55 **Genetic algorithms in engineering systems** A.M.S. Zalzala and P.J. Fleming (Editors)

Volume 56 **Symbolic methods in control system analysis and design** N. Munro (Editor)

Volume 57 **Flight control systems** R.W. Pratt (Editor)

Volume 58 **Power-plant control and instrumentation** D. Lindsley

Volume 59 **Modelling control systems using IEC 61499** R. Lewis

Volume 60 **People in control: human factors in control room design** J. Noyes and M. Bransby (Editors)

Volume 61 **Nonlinear predictive control: theory and practice** B. Kouvaritakis and M. Cannon (Editors)

Volume 62 **Active sound and vibration control** M.O. Tokhi and S.M. Veres

Volume 63 **Stepping motors: a guide to theory and practice, 4th edition** P.P. Acarnley

Volume 64 **Control theory, 2nd edition** J.R. Leigh

Volume 65 **Modelling and parameter estimation of dynamic systems** J.R. Raol, G. Girija and J. Singh

Volume 66 **Variable structure systems: from principles to implementation** A. Sabanovic, L. Fridman and S. Spurgeon (Editors)

Volume 67 **Motion vision: design of compact motion sensing solution for autonomous systems** J. Kolodko and L. Vlacic

Volume 68 **Flexible robot manipulators: modelling, simulation and control** M.O. Tokhi and A.K.M. Azad (Editors)

Volume 69 **Unmanned marine vehicles** G. Roberts and R. Sutton (Editors)

Volume 70 **Intelligent control systems using computational intelligence techniques** A. Ruano (Editor)

Adaptive Prediction and Predictive Control

P.P. Kanjilal

The Institution of Engineering and Technology

Published by The Institution of Engineering and Technology, London, United Kingdom

First published 1995
Reprinted 2008

The Institution of Engineering and Technology
Michael Faraday House
Six Hills Way, Stevenage
Herts, SG1 2AY, United Kingdom

www.theiet.org

While the author and the publishers believe that the information and guidance given in this work are correct, all parties must rely upon their own skill and judgement when making use of them. Neither the author nor the publishers assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

British Library Cataloguing in Publication Data
A CIP catalogue record for this product is available from the British Library

ISBN (10 digit) 0 86341 193 2
ISBN (13 digit) 978-0-86341-193-9

Printed in the UK by Short Run Press Ltd, Exeter
Reprinted in the UK by Lightning Source UK Ltd, Milton Keynes

To my parents
Bina and Gopal Chandra Kanjilal,
to my wife
Sagarika,
and to our children
Debayan and Sreyashi

CONTENTS

APPENDICES

PREFACE

Introduction

This book is about prediction and control of processes expressed by discrete-time models. It is assumed that the characteristics of the process may vary with time. The processes concerned may be linear or nonlinear, periodic or nonperiodic, single-input single-output or multi-input multi-output or simply output-only time series.

In this book, the prime emphasis is on adaptive prediction. This is a field which is of interest to practitioners and researchers belonging to various disciplines. For the same objectives, usually different approaches are used by different groups, sometimes unaware of alternative methods or unaware of implications as viewed by analysts of other disciplines. The prime aim of this book is to provide a unified and comprehensive coverage of the principles, perspectives and methods of adaptive prediction. One special feature of this book is the inclusion of a number of prediction methods, which are potent but are either new or are yet to be widely used.

Control often follows predictions. Adaptive control is a more cohesive discipline than adaptive prediction. Again, within adaptive control, the predictive control classes are of particular interest because of their inherent robustness and implementability in difficult real-life situations. These control methods are based on predictions or specified predictive performances. This book presents an introductory exposure to the popular methods of predictive control.

The numerical and computational aspects of the prediction and control methods used often influence their success in applications, and hence have been given due consideration as far as possible.

This book is intended to be of use to students, researchers, practitioners, as well as to nonexperts. It can form a one semester course for graduate classes or may be selectively used for undergraduate classes. Complex mathematical symbols or expressions are avoided, and efforts have been made to ensure that lack of mathematical expertise does not hinder the readers' comprehension of the subjects treated. So nonspecialists should also find this book readable and understandable. At the same time, the rigour required for the proper exposure of the subjects has not been compromised.

There are 14 chapters in this book. The introductory and background subjects are presented in the first three chapters, which include discussions on process modelling, selection of models, and estimation of parameters. Chapter 4 provides an exposure to the popularly used methods of prediction. Chapters 5 to 11 discuss various methods of adaptive prediction for linear and nonlinear processes. In brief, the studies include input-output model based predictions, Kalman filter and state-space predictors, orthogonal transformation based predictors, and predictors based on hierarchical models including the Group Method of Data Handling and neural networks. Chapters 12 and 13 are devoted to predictive control; the study includes the input-output model based long range predictive control and the state-space model based method of linear quadratic control. Chapter 14 discusses the concept of extraction of information through smoothing and filtering of the data.

The book is largely self-contained, although conventional or widely studied topics are briefly dealt with; wherever possible newer features have been introduced and related interpretations have been added. The reader is expected to find that at least some of the subjects presented in Chapters 3, 7, 10, 11, and 14 are new.

The theoretical discussions are supported by a large number of examples, application studies and case studies selected from diverse areas. The supportive appendices provide background materials, implementation ideas for the algorithms, computer programs etc., which are designed to help the readers' understanding and to ease the efforts in implementing the presented ideas.

This book is also supported by a floppy disc, for which the author may be directly contacted.

The subject of this book is extensive, and newer facets are always appearing; this book is expected to provide a broad introductory coverage. In spite of all efforts, there may be omissions, errors or imprecisions in expressions; the readers are requested to convey their criticisms and suggestions for improvement, which will be most thankfully received by the author.

Acknowledgement

During the long preparatory period of this book, many people helped in many ways in its preparation; I apologize for not being able to mention all their names but I extend my most sincere thanks to all of them.

It is a pleasure to thankfully acknowledge the inspiration and the suggestions received from Prof. G. P. Rao. I am very thankful to my present and past Ph.D. students, Goutam Saha and Sarbani Palit for their help and computational support. The different chapters of this book were reviewed by my colleagues Amit Patra, Sidhartha Mukhopadhyay, Saswat Chakrabarty, Sarit Das, and Anjan Sarkar; I am extremely grateful for their thoughtful comments and suggestions. I am also thankful to Kumar Deepak, A.V.B. Subrahmanyam, Prodyot Dey, Debarag Banerjee, and to other students who read parts of the manuscript and gave their valuable observations.

Over the period of the writing of the book, many people were involved at different stages with the typing of the manuscript, and preparation of illustrations etc. I am very grateful to all of them. Thanks are particularly due to Subhas Giri, Thomas Koickal, Kullu S. Rao, Muktiram Bose, and Ayan Banerjee.

I am extremely thankful to my publisher the Institution of Electrical Engineers for their understanding, and for permitting me the long preparatory period for this book.

Finally I am most grateful to my wife Sagarika who has been almost the nonwriting coauthor of this book; without her constant inspiration and support and immense sacrifice this book could not have been completed. I am also indebted to our small children Debayan and Sreyashi, for the loss of the pleasant time that we could have spent together. My parents have been a constant source of inspiration, and encouragement, which always helped the best that I have been able to produce.

Department of Electronics and ECE
I.I.T. Kharagpur, 721-302, India.

Partha Pratim Kanjilal

CHAPTER 1

INTRODUCTION

Meaningful predictions, control based on predictive performance, and robust implementation are the main themes of this book.

The objective of a prediction exercise is to determine the future values of a variable based on the available information. The more representative the information, the better is the chance for producing close predictions. As the golden rule says that there is no golden rule, it can also be said that the best prediction is that the prediction cannot be the best. This is because, in real life, *bestness* cannot be precisely defined. Predictions depend on the data or measurements available, the system generating the data, the environment influencing the measurements, the dynamic state of the system, and the prior subjective knowledge about the process etc.; there is a possibility of implicit inaccuracy or imprecision with each of these. So the sensible objective will be to generate *meaningful predictions*, which is the prime subject of this book.

If the process is controllable, the knowledge of the predictive performance can be used in the design of control laws for the process in order to drive its output to the desired set point. If the predictions are dependable, the control methods have a better chance of reaching the targets in the expected time. Broadly speaking, the class of control methods, which incorporate information or assumptions pertaining to the future are referred to as *predictive control*, a subject which also forms a part of this book.

The Prediction Problem

The fundamental problem concerning prediction is the mathematical modelling of the process from the available information; the model is used to generate the predictions. The basic issues involved are as follows.

(a) Representativeness of the data

The observations or measurements obtained from the process may be contaminated with noise, because the variable concerned may not be directly or precisely measurable.

For example, consider the measurement of hardness of coal; one method is to collect bulk samples, which are then subjected to tumbling; the resulting degree of granulation leads to the computation of the hardness of coal. So it is important to appreciate that the variable of interest may not be the actual measurement but some underlying process value.

Again consider the case of measurement of furnace temperature using a thermocouple. Here an accuracy of measurement typically beyond the first decimal place in degrees centigrade is not usually expected.

So the modelling exercise should incorporate the consideration of the true representativeness of the data. Besides the observed data, prior subjective knowledge about the process can greatly help formulate reliable models.

(b) Statistical characterization of the data

Often, to estimate the parameters of the model, the data are assumed to possess certain idealistic statistical properties, which may be only loosely true, e.g., the assumption that the noise associated with a real-life data series being white Gaussian. Also, if the data length is not *long enough* (a term, which again is not well-defined, a typical figure being 50 to 100 data points), the statistical characterization of the data may not be dependable.

(c) Modelling versus prediction

A model, which fits the data well, may not necessarily be representative, and hence may not produce sensible predictions. In fact with increased model order, it is always possible to get a closer fit with the data. The model fit against a separate block of data, not used for developing the model, will be a better indication of the validity of the model. If the model is valid irrespective of the choice of the data within the complete set, it is expected to be able to produce close predictions.

(d) Rate of adaptation

Real life is not static. Hence, as time progresses, adaptation or modification of the parameters of the model will be necessary. The rate of adaptation depends on the degree of dynamics in the underlying process. The validity of prediction requires the model to be representative and also to remain representative over the period of prediction.

(e) Prediction and its validity
Once a model is developed, the model and the available subjective knowledge may be used to produce predictions. There being no standard method of testing the validity of predictions, any validity test should incorporate the *overall sense* in the predictions. For example consider the mean square error (MSE) criterion:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2,$$

where $\hat{y}$ is the predicted value of y. The deviation of the prediction at one point, which may be an outlier from an erroneous measurement, can make the cumulative square error too large irrespective of the predictions being otherwise sensible. So although mean square error is an informative index, it cannot be the only decisive factor for the quality of prediction. In fact a *plot* of the predictions along with the observations will provide a useful insight into the closeness of prediction.

(f) Implementation aspects
Some of the important considerations from an implementation point of view are as follows.

i) The numerical robustness (e.g., singular values are more robust than eigenvalues, numerically)
ii) the computational stability (e.g., UD updating through square-root filtering of the error covariance is computationally more stable than the direct covariance matrix updating in recursive least squares estimations)
iii) the real-timeliness property (e.g., low-pass filtering introduces phase-shift, thereby damaging real-timeliness).

In practice, once the user is familiar with the complexities involved with modelling and prediction, necessary steps may be taken in the choice of the process information, the algorithms and in the computations such that sensible predictions are produced.

The modelling and prediction problem is viewed differently by people belonging to different disciplines, who have their own preferences for suitable methods. Some concepts are similar; for example, the central moving average used by time series analysts is similar to fixed interval smoothing used by systems analysts or the bidirectional filtering used in the field of signal processing. On the other hand some practices are dissimilar; for example, often one tries to estimate the constant or average term like a_o in

$$y(k) = a_o + a_1y(k-1) + a_2y(k-1) + e(k),$$

where y is the output variable, a_o, a_1, a_2 are the parameters and e is the noise term; a control engineer will avoid such a practice, owing to identifiability problems. This is because, for sensible estimation, some amount of dynamism is necessary in the data. Ideally the data need to be persistently exciting. The estimation is difficult here, as the data associated with a_o remain static, at unity.

In this book, some efforts are made to compile and combine concepts and methods popular with different disciplines.

The control problem

Compared with prediction, the control problem is more clearly defined, because the objective is to drive the process output or a particular variable to a specified set point. There are two basic issues:
(i) *Identification* of the process which involves the modelling and the parameter estimation problems.
(ii) *Control calculation* which involves computation of the control input optimizing a predefined cost criterion.

While proper identification of a dynamic process is still a difficult problem, the adaptive control problem is comparatively well defined, and closed form solutions are possible, which can also absorb imprecision in identification to some extent.

In this book, a special class of control methods known as *predictive control* is studied.

The predictive control strategy incorporates predictions of the controlled variable or assumptions relating to the future values of parameters of the controller. The basic objective is to enhance robustness of the controller. In general, robustness and speed of response are two contradictory demands on a controller. In practice, however, there is greater emphasis on robustness, which is achieved through the knowledge of the predictive controller, although at the cost of speed of response.

In the present text two broad categories of predictive control are studied; these are based on input-output or transfer function models and on state-space models. The former is referred to as the *Long range predictive control*, where the present time is used as the reference point from which predictions are computed, and a control law is determined to optimize the performance in the future. In the case of state-space model based predictive control (also known as Lineâr Quadratic (LQ) control, as a quadratic cost is minimized and a linear model of the process is considered), the terminal point in the future is used as the reference; the present control is determined so that the terminal conditions are satisfied.

Organization of the book

The subject matter presented in this book is arranged into five broad groups:

(1) Preparatory studies: Chapters 2 and 3.
(2) Prediction methods and applications for linear models: Chapters 4 to 7.
(3) Prediction methods and applications for nonlinear models: Chapters 8 to 11.
(4) Predictive control studies: Chapters 12 and 13.
(5) Smoothing and filtering aspects: Chapter 14.

The preparatory studies presented in Chapters 2 and 3 overview the possible types of process models and the common methods for parameter estimation respectively. The models discussed are the transfer-function models based on the input-output data, the models developed from frequency domain characterizations, and the structural models comprising the trend, the periodic component(s) and the random component. In real life most processes are better represented by stochastic models incorporating additional random components; so knowledge of stochastic processes, also featured in Chapter 2, is a prerequisite of the modelling and estimation studies.

The parameter estimation studies presented in Chapter 3 mainly concentrate on the method of the least squares. Particular emphasis has been laid on robust implementation through singular value decomposition. The problems of selection of the optimal model, and the assessment of the validity of the model are also addressed.

Chapter 4 presents some of the well studied methods of prediction which are popular with time series analysts and statisticians. The topics discussed include the exponential smoothing based methods and the Box and Jenkins methods.

Chapter 5 presents the input-output model based predictors. The concept of constraining the prediction increments is introduced, which incorporates prior subjective knowledge about the underlying process into the predictor, and is expected to lead to robust prediction.

The state-space modelling ideas are introduced in Chapter 6. The main attraction of a state-space approach is that the measured process variables as well as internal process variables which cannot be accessed can be incorporated in the model; a vast amount of thoroughly researched results are also available to the designer. A detailed study on the Kalman filter, the optimal linear filter, is included. The optimal state-space predictor is derived from the Kalman filter.

Chapter 7 introduces the concepts of orthogonal transformation and studies the two particular types of transformations: the singular value decomposition (SVD) and the Walsh Hadamard transformation (WHT); while the former is exceptionally robust numerically, the latter is extremely simple from a computational point of view. SVD which has so far rarely been used in prediction applications, features again and again in the present text. For example SVD is used as an algebraic tool for matrix operations in Chapter 3, for the modelling and prediction of periodic series, quasiperiodic series and nonlinear input-output processes in Chapters 8 to 11, and used for filtering in Chapter 14. The application of SVD and WHT for modelling and prediction of nearly periodic time series is explored in Chapter 7.

Most real life processes are nonlinear to varying extents. Chapter 8 starts with a discussion on the basic features characterizing nonlinearity of a process, which is followed by assessment of nonlinear periodicity through state-space diagrams. A review of some of the popular methods of modelling and prediction of nonlinear time series is also presented.

Chapters 9 and 10 concern hierarchical models for nonlinear time series or input-output processes. The Group Method of Data Handling (GMDH) presented in Chapter 9 and the neural network models discussed in Chapter 10 are both powerful methods. In both the cases, multiple stages of simple nonlinearity are used to develop complex nonlinear models. It has been shown that in both the cases application of SVD and QR with column pivoting (QRcp) factorization can lead to parsimonious designs with improved performance. The neural-net models are comparatively more versatile and powerful. The parameters of the network may be estimated using a nonlinear optimization method. Applications with the network operating both with time-domain data and (orthogonally) transformed data are presented.

Chapter 11 is devoted to the modelling and prediction of quasiperiodic series using SVD. Two basic approaches are explored. In the first approach a quasiperiodic series is decomposed into relatively periodic component series (having the same period length) in hierarchical levels of nonlinearly transformed spaces; the periodic components are individually modelled linearly or modelled using neural networks. In the other approach, the series is decomposed into multiple periodic components (having different period lengths) in the time domain; each periodic component is expressed by a linear model. Both the approaches can produce prediction of one complete period.

The information conveyed through prediction is often used for control either directly or indirectly. If the process concerned can be controlled, predictive information or assumptions can be directly used in the design of predictive controllers. Two broad categories of predictive controllers are considered, namely, the input-output model based controllers and the state-space model based controllers, which are discussed in Chapters 12 and 13 respectively. The predictive information is incorporated in these two approaches differently. In the input-output model approach, 1 to multistep prediction is used with the present time as the point of reference; in the case of state-space approach, a terminal point in future is used as the reference.

Chapter 14 is devoted to the smoothing and filtering studies which are particularly important for prediction and control problems. The property of the real-timeliness of the data has been given due consideration. The problems addressed include low-pass filtering without phase-shifts, extraction of signals from noise corrupted data, and estimation of pattern for nearly repetitive processes.

Lastly a number of Appendices are included which support the subjects covered in the various Chapters; it is hoped that the Appendices will aid the comprehension of the readers as well as assist implementational efforts.

Special Features

In the widely studied field of adaptive prediction and control, this book presents potential concepts, and reviews newly developed methods for the modelling and prediction of nearly periodic and quasiperiodic series and complex input-output processes as well as numerically robust and computationally efficient implementation ideas for almost all the presented methods of prediction and control. In addition, efforts are made to bring together the concepts and practices popular with diverse disciplines. The studies presented incorporate due consideration of real life problems, which have been explored using illustrative examples and case studies.

Bibliography

Remarks: The areas covered in this book broadly feature in the following texts.

[1] Aström, K.J., and B. Wittenmark (1988): *Adaptive Control*, Addison Wesley, Reading, Mass.

[2] Bierman, G.J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[3] Box, G.E.P., and G.M. Jenkins (1976): *Time Series Analysis, Forecasting and Control*, Holden-Day, San Francisco.

[4] Draper, N. R., and H. Smith (1981): *Applied Regression Analysis*, 2nd edn., Wiley, New York.

[5] Golub, G.H., and C.F. Van Loan (1989): *Matrix Computations*, 2nd edn., The Johns Hopkins Univ. Press, Baltimore.

[6] Kashyap, R.L., and A.R. Rao (1976): *Dynamic Stochastic Models from Empirical Data*, Academic Press, New York.

[7] Makridakis, S., S.C. Wheelwright, and V.E. McGee (1983): *Forecasting Methods and Application*, 2nd edn., Wiley, New York.

CHAPTER 2

PROCESS MODELS

Modelling concerns the mathematical representation of the nature of the process with respect to its environment; the purpose of modelling and the type of data available are important considerations.

2.1 INTRODUCTION

The understanding and study of any process, requires a mathematical representation or model of the process. The process may be an input-output process, or a time series (i.e. an apparent output only process). The model is based on the prior physical or subjective knowledge about the process, the measured data on the inputs and the outputs of the process, and the physical and engineering laws governing the working of the process. The primary requirements of a model are (i) representativeness and (ii) long-term validity.

If the model is a complete and exact representation of the process, it is called a *deterministic* model, and the process is called a deterministic process. The parameters of such a model are precisely known, and the model can be used to produce exact prediction of the process response from the past data. However, most real-life processes cannot be represented by deterministic models, because of the dynamic nature of the process and the noise (meaning lack of information) and other uncertainties being associated with the available data; so, the description of the process by the model can only be probabilistically close to the actual process. A model which incorporates noise or disturbance terms to account for such imprecision in the knowledge of the process is called a *stochastic* model.

Modelling involves selection of the process variables to be considered, selection of the class of model, selection of the model structure, estimation of the parameters of the model and testing of the validity of the model. There has to be economisation in the degree of complexity of the model, as otherwise the overall validity of the model tends to suffer; it is desirable that the model is as simple as practicable. If the characteristics of the process change with time, the parameters of the model are to be estimated recursively; such a model is referred to as an *adaptive* model.

Throughout this book, models of different categories have been used for various types of applications, an introductory summary of which is presented in Sec.2.2. This is followed by a study on stochastic processes in Sec.2.3; knowledge of stochastic processes is important, because most real-life processes are stochastic in nature. Next, two important classes of models are discussed. The transfer function models are studied in Sec.2.4, and the models based on frequency domain analysis are described in Sec.2.5, where data sampling aspects of measurements are also discussed. The structural properties of processes like the trend and the seasonality etc. can be used for configuring a model, as discussed in the context of structural modelling in Sec.2.6. Structural modelling through periodic decomposition has also been introduced.

2.2 PROCESS MODELS AND THEIR CHOICE

For representative modelling, the choice of the model, the estimation of parameters and the testing for the validity of the model are all equally important. Parameter estimation and model validation aspects are discussed in Chapter 3; this section discusses the candidate classes of models and the considerations leading to the choice of the models.

2.2.1 Classes of Models

Five broad classes of discrete-time models are discussed; these are

(1) time series and transfer-function models,
(2) models based on trigonometric functions,
(3) state-space models,
(4) models based on orthogonal transformations,
(5) hierarchical models including GMDH and neural networks.

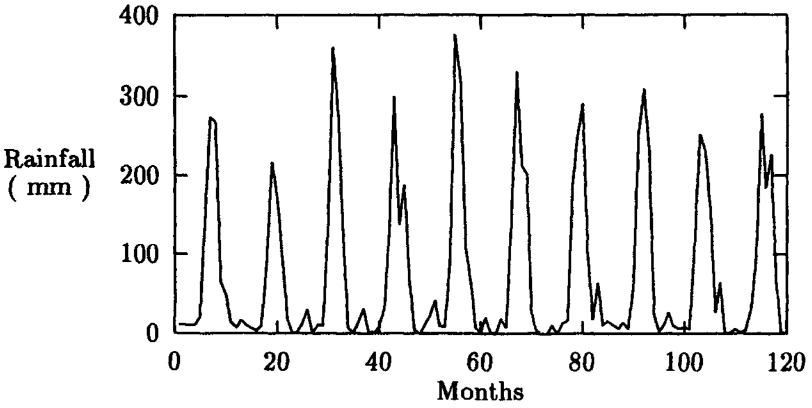


Figure 2.2.1 Monthly rainfall pattern in India over the period 1940 to 1949 (Appendix 7F).

A common feature of these models is that they can all accommodate a certain degree of uncertainty, and can adapt to time-varying process dynamics. The modelling of some processes may require incorporating features of more than one class. An outline of the stated classes of models follows, and is summarized in Table 2.2.1.

Time series and transfer-function models

Time series is a sequence of observations on a variable of the process. It may or may not have a periodic component associated with it; typical examples are the monthly rainfall pattern over the years (Fig.2.2.1), and the variations in the rate of rotation of the earth (Fig.2.2.2).

Time series may be represented by AR, IAR, ARMA, ARIMA etc. type models, which are based on polynomial operators in discrete time; these models are discussed in Sec.2.4. If a time series shows structural features (like trend and periodicity), the structural components may be separately modelled. Structural modelling features in Sec.2.6.

Remarks

(a) Here as well as elsewhere in this book, a *time series* has often been referred to as a 'process'. (b) The reference to 'time' in time series is not a limitation. The studies on time series also apply to sequences in *space*.

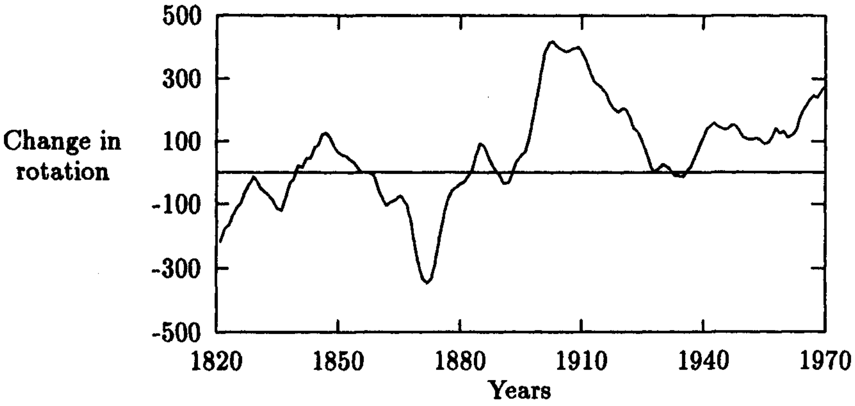


Figure 2.2.2 The yearly variations in the rotation rate of the earth in units of 10^{-5} seconds (Appendix 8B).

Transfer-function models are natural extensions of time series models. It is expected that the process in question is subjected to certain external inputs, which influence the output of the process; for example, the temperature (output) of a furnace varies with the change in the fuel-gas flow (input) into the furnace. Transfer-function models have additional terms for the exogeneous input(s), as in the ARMAX (that is ARMA with eXogeneous input(s)), and ARIMAX models; these models are discussed in Sec.2.4.

The transfer-function models have been widely used in this book.

Models based on trigonometric functions

Processes with regular or irregular periodicity can be analysed in frequency domain and can be modelled in terms of components, expressed as trigonometric functions. Besides modelling, frequency domain characterization provides useful information in the design of filters as well as in assessing the appropriate rate of sampling of continuous-time signals for discrete time modelling. A detailed study of frequency domain analysis and modelling based on trigonometric functions is in Sec.2.5.

State-space models

State-space models have the unique feature that along with variables which are known or can be measured, the variables which are internal to the process and cannot be measured are also incorporated into the model; this is why a state-space model is also called an *internal model*, whereas a model based on measurable variables is called an *external model*. For example, consider the problem of modelling the internal temperature variations of steel ingots (discussed in Sec.6.8) while being heated inside a soaking pit (or furnace) before rolling. Here only the temperature inside the furnace is measurable, whereas the ingot surface temperature and the ingot-centre temperature are not measurable; all the three variables are considered as state variables in the state-space model of the furnace. The fuel-gas flow to the furnace is regarded as the exogeneous input in the model.

Any transfer-function or time series model can have a state-space representation but the converse is not true. Processes with or without periodicity can be modelled by state-space models. Chapter 6 is devoted to the study of state-space models and their applications. State-space formulation of LQ control features in Chapter 13. The state-space models for optimal smoothing is presented in Sec.14.2.

Models based on orthogonal transformations

In this book there is particular emphasis on the use of singular value decomposition (SVD) for modelling. The special feature of SVD is that it results in optimal compaction of information (as discussed in Sec.7.6).

The models based on SVD are particularly suitable for time series which are nearly periodic (for example see Fig.2.2.1), or quasiperiodic in nature (for example the yearly averaged sunspot series as shown in Fig.2.2.3).

The principle of modelling for the nearly periodic series is that the consecutive periods are aligned into consecutive rows of a matrix, which is SV-decomposed; the decomposed components are now modelled, typically as a time series. This subject is treated in detail in Secs.7.7–7.8.

A quasiperiodic series, can be decomposed into components which are individually nearly periodic, and hence can be modelled the same way as above. Such a modelling scheme features in Sec.11.4.

The two attractive features of SVD based modelling are

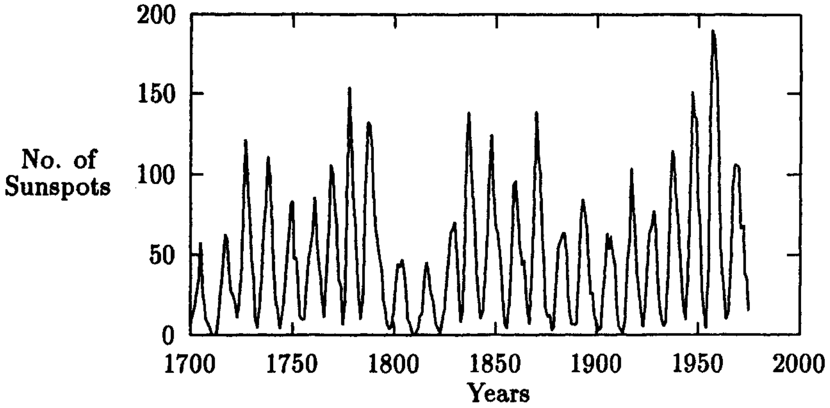


Figure 2.2.3 The series of yearly averaged sunspot numbers (Appendix 8A).

that (i) one or multiple *period* ahead prediction may be produced, and (ii) SVD, which is extremely robust numerically, ascribes robustness to the model.

Hierarchical or multilayer models

These models are primarily suitable for time series and input-output processes with nonlinearity; quasiperiodic processes can also be modelled. The three types of models studied are:

(1) models based on Group Method of Data Handling (GMDH),

(2) neural network models, and

(3) models based on singular value decomposition with or without nonlinear transformation.

All these models have hierarchical stages or layers, where each stage incorporates simple elements of nonlinearity. Since most processes contain a certain degree of nonlinearity, these models are applicable for nonlinear as well as nearly linear (and nearly periodic) processes.

Typical processes that can be modelled are multi-input single output processes like the economic inflation process, quasiperiodic processes like the yearly averaged sunspot series etc. Nearly periodic processes like the homogeneous monthly rainfall series may also be modelled using such hierarchical models. Chapters 9, 10 and 11 are devoted to the study of hierarchical models.

Table 2.2.1 Summary of the main classes of models studied

Model type	Type of process modelled	Features and applications	Reference
State-space models (based on state variables, input and output variables and noise)	Processes with or without periodicity, linear or nonlinear processes, output only or input-output processes.	Internal models; variables modelled may or may not be measurable. p-step ahead prediction, LQ-control smoothing and filtering.	Ch.6 Ch.6, Ch.13, Ch.14
Orthogonal transformation (Singular value decomposition) based models	Periodic processes modelled using SVD. Quasiperiodic processes modelled using a) SVD and nonlinear transformation with linear modelling, b) SVD and nonlinear transformation with nonlinear modelling, c) SVD and multiple pattern decomposition with linear modelling.	Periodic modelling, one period or one pseudo-period ahead prediction, pattern decomposition and pattern extraction, smoothing and filtering.	Ch.7 Ch.11 Ch.11, Ch.14 Ch.14
Time series/ transfer-function models	data sequence or output only process/ Input-output process with or without periodicity.	p-step ahead prediction, constrained prediction, predictive control of input-output processes.	Ch.4,5 Ch.5 Ch.12
Frequency domain analysis based models	Processes with periodicity.	Signal analysis, modelling and prediction, filtering.	Ch.2,6
Multi-layer models (GMDH and Neural Network models)	Periodic and quasi-periodic processes with nonlinearity, input-output processes.	p-step ahead prediction, one period ahead prediction.	Ch.9 Ch.10 Ch.11

Remarks

(a) In all the cases, efforts are made to develop parsimonious models, i.e. only the essential number of variables are to be included in the model, and the model

order is kept as low as possible.
(b) The degree of accuracy of the data should be duly considered. In case of real-life applications, the model needs to be protected against irrelevant information influencing the model.
(c) The underlying assumptions in building the model as well as the limitations of the model should be clearly stated and verified.

2.2.2 Choice of models

Three basic attributes of a process which influence the choice of the type of model are linearity, periodicity, and stationarity.

Linearity: No process is exactly linear, but acceptable solutions may be obtained by considering the process piece-wise linear, or locally linear (that is linear around the operating point). In case of nonlinearity, efforts may be made to linearize the representation of the process through nonlinear transformation. If linearization is not possible, or if it is thought that a nonlinear model will be more representative, a nonlinear model is used. The preference for a linear model is because (a) it is computationally simpler, (b) the desired statistical features for nonlinear models are not fully established.

Periodicity: Some processes show periodic variations. A periodic process is characterized by three components: the length of the period, the pattern which repeats over the periods, and the relative magnitude of the patterns. When the length of the period as well as the pattern remains almost unchanged a linear model may be produced. If both or either of the period length and the pattern vary, the process is called a quasiperiodic process, and the process can be modelled as a nonlinear periodic process or as a combination of periodic processes (see Sec.8.3 and Chapter 11).

Stationarity: This is the property of the process characteristics remaining unchanged with time (or space). If the process is not stationary, its degree of nonstationarity may be decreased by suitable transformations (e.g., logarithmic transformation). A relatively less stationary process will require more frequent adaptation. The characteristics of stationary processes are discussed in Sec.2.3.1.

An exhaustive study of the possible classes of models and the criteria for their choice is beyond the scope of this book. The main considerations are:
(a) whether a periodic or quasiperiodic model is desired, or whether the model is to be nonperiodic in nature,
(b) whether a linear or a nonlinear model is to be designed,
(c) whether one or more variables of interest are inaccessible or unmeasurable, etc.
(d) whether the model is to be deterministic or stochastic in nature.

Remarks
(a) The modelling exercise is greatly simplified if the process is linear or periodic and stationary. In real life, most processes are not so, and efforts are made to preprocess the data to increase the degree of stationarity, periodicity or linearity prior to modelling. The preprocessing can involve differencing of the data, nonlinear transformations (Sec.8.2.3), smoothing or filtering (discussed in Chapter 14), orthogonal transformation (discussed in Chapter 7) or other linear transformations.
(b) Between different contending classes of models, the choice will depend on the purpose of modelling, the quality of data available, and the computational preferences of the user.

2.3 STOCHASTIC PROCESSES

The word *stochastic* is of Greek origin, meaning *to guess*. A stochastic process is a random process evolving in time whose behaviour can be analysed statistically but cannot be predicted precisely.

The uncertainty or the unknown disturbance in the observed process is described by a stochastic process and because of the presence of this uncertainty, the overall process is also called a stochastic process.

The more accurate the statistical representation of the uncertainty in the process model, the higher the probability is of the model response being close to the actual response of the process. Hence it is necessary to study the stochastic nature of the disturbance processes.

2.3.1 Basic Concepts and Processes

Some of the basic concepts and some of the basic types of processes are discussed here.

Stochastic processes

A stochastic process X(t) belongs to the family of random variables $\{x(t),\ t\in T\}$, where the random variables are indexed with the parameter t, all of whose values lie in an index set T. The parameter t is usually interpreted as time (although this is not a limitation). The random variables may be scalars or vectors.

With the notion of time as the index, if the time index set is defined as $T = \{t: 0<t<\infty\}$ or $T = \{t: t\geq t_0\}$, the process is called a continuous-time stochastic process. On the other hand, if T is a set of discrete time instants, $T = \{k+i: i = \ldots, -1, 0, 1,\ldots\}$, the process x(k) is called the discrete time stochatic process. The sampling period is chosen as time units which are distinct but not necessarily equispaced.

Stationary processes

A stochastic process x(k) is said to be strict-sense stationary, if its probability density functions are independent of the shift in the time origin, that is if the two processes x(k) and $x(k+\tau)$ have the same probability density function for any τ, they constitute a strict-sense stationary process.

The processes x(k) and y(k) are jointly stationary, if the joint statistics of x(k), y(k) are time-invariant, i.e. the joint statistics are the same as that of $x(k+\tau)$, $y(k+\tau)$, for any τ.

The process x(k) is said to be wide-sense (or weak-sense) stationary, if its mean value is constant and its autocorrelation *R* depends only upon the time difference, expressed as

$$E\{x(k)\} = \bar{x}, \text{ a constant,} \tag{2.3.1a}$$

$$E\{x(k+\tau)x(k)\} = R(\tau). \tag{2.3.1b}$$

Wide-sense stationarity does not require probability density function to be invariant unlike the strict-sense stationary process.

The stationary processes for which time averages equal probabilistic averages, (i.e. ensemble average or expectation) are called ergodic. Not all stationary processes are ergodic.

If the discrete time process {x(k), k = 1,2,...,N} is ergodic, it can be completely described by its expectation

$$\bar{x} = E\{x(k)\} = \lim_{N \longrightarrow \infty} \frac{1}{N} \sum_{k=1}^{N} x(k),$$

and by its autocorrelation function

$$R(\tau) = E\{x(k)x(k+\tau)\} = \lim_{N \longrightarrow \infty} \frac{1}{N} \sum_{k=1}^{N} x(k)x(k+\tau).$$

Remark: A covariance stationary process {x(k)} may be decomposed as $x(k) = x_d(k) + x_n(k)$, where x_d is a purely deterministic component, and x_n is a purely stochastic component; such a decomposition is known as *Wold's decomposition* (Wold, 1954).

Markov processes

The Markov property states that the future depends on the knowledge of the present, and not on the knowledge of the past. A stochastic process {x(k), k∈T} is called a *Markov process*, if

$$P\{x(k+1)\,|\,x(k),x(k-1),\ldots,x(k_o)\} = P\{x(k+1)\,|\,x(k)\}; \qquad (2.3.2)$$

in other words, the conditional (or transitional) probability density function for x(k+1) depends only on its present value x(k) and not on any value in the past; here the time instants $k_o<\ldots<(k-1)<k<(k+1)$ belong to the set T. Thus the concept of *probabilistic causality* (see Appendix 12) is inherent with the Markov process.

The joint probability density function of the Markov sequence (scalar or vector x(k+1), x(k), x(k-1), ..., $x(k_o)$) is completely specified in terms of the initial probability density function $P\{x(k_o)\}$ and the transition density function $P\{x(k+1)\,|\,x(k)\}$. Following (2.3.2), the joint probability density function is given by

$$P\{x(k+1),x(k),x(k-1),..,x(k_o)\}$$
$$= P\{x(k+1)\,|\,x(k)\}P\{x(k)\,|\,x(k-1)\}...P\{x(k_o+1)\,|\,x(k_o)\}P\{x(k_o)\}.$$

The Markov process as defined above is sometimes called the *first order Markov sequence.* The underlying concept may be extended and a second order Markov process may be defined as the one requiring the two most recent information elements to describe the future, i.e. with $(k+1)> k >...>k_o$,

$$P\{x(k+1)\,|\,x(k),x(k-1),...,x(k_o)\} = P\{x(k+1)\,|\,x(k),x(k-1)\}. \tag{2.3.3}$$

Higher order Markov processes may also be defined the same way.

Remarks

(a) A subset of a Markov sequence is also a Markov sequence.

(b) If a Markov sequence is reversed in time, it still retains Markov property:

$$P\{x(k+1)\,|\,x(k+2)...x(k+n)\} = P\{x(k+1)\,|\,x(k+2)\},$$

where $(k+1)<(k+2)<...<(k+n)$; all belong to the set T.

Gaussian distribution

A random variable x is called a Gaussian or normally distributed random scalar variable with mean$\{x\} = \bar{x}$, and covariance, $\text{cov}\{(x-\bar{x})\} = \sigma^2$, if its probability density function is given by

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right). \tag{2.3.4}$$

The expression 'x is $N(\bar{x},\sigma^2)$' means, x is Gaussian (normal) with mean $\bar{x}$ and variance σ^2.

A random vector is called a Gaussian or normally distributed random n-vector $x = [x_1, x_2,..., x_n]^T$, if its probability density function is given by

$$P(x) = \frac{1}{(2\pi)^{n/2}}\frac{1}{(\det \mathbf{P})^{1/2}}\exp[-\tfrac{1}{2}(x-\bar{x})^T\mathbf{P}^{-1}(x-\bar{x})], \tag{2.3.5}$$

where the mean vector $\bar{x}$ and the covariance matrix $\mathbf{P}$ are given by

$$\bar{x} = E\{x\}, \qquad \mathbf{P} = E\{(x-\bar{x})(x-\bar{x})^T\}.$$

Thus the mean and the covariance uniquely describe the Gaussian probability density function.

A stochastic process $\{x(k), k \in T\}$ is called a Gaussian or *normal process*, if for any n points: $k_1, k_2, \ldots, k_n$ in T, the random variables,

$$x(k_1), \ldots, x(k_n), \tag{2.3.6}$$

are jointly of Gaussian or normal distribution. The probability law for the joint density function of the Gaussian stochastic process (2.3.6) is completely specified by the two positive parameters, the mean and the autocorrelation function, given by

$$E\{x(k)\} = \bar{x}(k), \qquad \text{for all } k \in T,$$

$$E\{x(k_1)x(k_2)\} = R(k_1,k_2), \qquad \text{for all } (k_1,k_2) \in T.$$

Remarks

(1) If $\mathbf{P}$, the covariance matrix of $\mathbf{x}$ in (2.3.5), is singular, the Gaussian property of $\mathbf{x}$ cannot be defined by the probability density function; the characteristic function can be used in that case.

(2) A stochastic process is called a Gauss-Markov sequence, if it has Gaussian distribution and at the same time is a Markov sequence.

(3) (a) A subset of a Gaussian vector is also Gaussian.
(b) Gaussian variables retain their Gaussian character under linear transformation.

White noise processes

A random sequence $\{x(k_n), x(k_{n-1}), \ldots, x(k_2), x(k_1)\}$ is said to be a purely random or white noise sequence, if $x(k_i)$ and $x(k_j)$ are completely independent for $i \neq j$. For such processes, the conditional density is the same as the marginal density.

$$P\{x(k_n) \mid x(k_{n-1})\} = P\{x(k_n)\}.$$

The implication is that (a) the white noise sequences do not possess any memory and (b) the present is independent of the past, while the future is independent of the present.

The autocorrelation function is given by

$$E\{x(k_i)x(k_j)\} = R\delta_{ij},$$

where δ_{ij} is the Kronecker delta function:

$$\delta_{ij} = \begin{cases} 1 \text{ for } & i = j, \\ 0 \text{ for } & i \neq j. \end{cases}$$

If the white noise process has a Gaussian distribution, it is called a *white Gaussian process*, which is completely specified by the mean and the covariance matrix. One widely studied white Gaussian noise process is the Brownian motion or Wiener process. Usually the white noise process is zero mean, and if stationary, its power spectrum will be constant.

Remark: Although in signal processing, the term *noise* implies 'lack of information or signal', *white noise* is an important entity. Because of its unique statistical properties, white noise is often used as the *input signal* for process identification.

2.3.2 Examples of Common Processes

Some of the commonly occurring stochastic processes are Brownian motion or Wiener process, Poisson process and Random walk. All these processes are characterized by being

(a) Markov processes and
(b) independent increment processes,

where an *independent increment process* $\{x(t)\}$ has the property that for $t_1<t_2<\ldots<t_n$, the differences,

$$(x(t_2)-x(t_1)),\ (x(t_3)-x(t_2)),\ldots,\ (x(t_n)-x(t_{n-1})),$$

are mutually independent.

Brownian motion (Wiener process)

A microscopic particle in a fluid moves in an erratic fashion due to random collisions with other particles and frictional resistance between collisions. This phenomenon is called *Brownian motion* after Botanist Robert Brown, who observed it in 1826. Brownian motion is also called the *Wiener process*, as this process was rigorously analysed by Nobert Wiener.

Any specified time interval r to s is expected to be much larger than the time between two collisions and hence the movement of the Brownian particle is the resultant of a larger number of random movements.

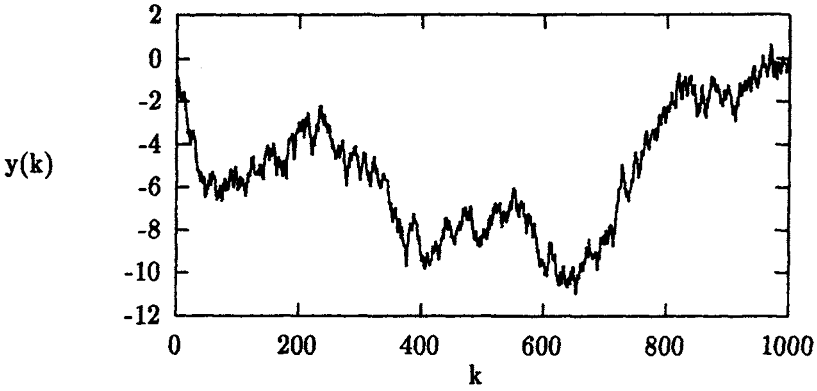


Figure 2.3.1 A typical Brownian motion series

The Wiener process is a stochastic process with stationary increments, $\{x(s)-x(r)\}$, which are normally distributed with zero mean and variance proportional to the time difference (s–r):

$$E\{(x(s)-x(r))\} = 0,$$

$$E\{(x(s)-x(r))^2\} = \sigma^2|s-r|,$$

where σ^2 is a positive constant. Thus the probability density of displacement from time r to s is the same as from time $(r+\tau)$ to $(s+\tau)$, since the density depends upon the length of the time interval and not on specific time reference.

Fig.2.3.1 shows a Brownian motion series $\{y(k)\}$ generated by passing a white Gaussian noise sequence $\{x(k)\}$ through an integrator, implemented as $x(k)/(1-q^{-1})$ (where q^{-1} is a unit discrete time backward shift operator, e.g., $q^{-1}y(k) = y(k-1)$):

$$y(k) = y(k-1) + x(k).$$

Poisson process

Poisson process x(t) is an integer valued stochastic process, constituting of (mostly nondecreasing) jumps of unit magnitude, occurring at random time intervals. $(x(t_1)-x(t_2))$ equals the number of events that occurred in the time interval (t_2, t_1). The probability of m

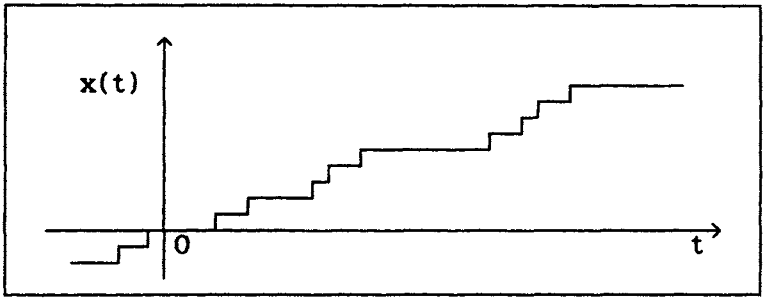


Figure 2.3.2 A Poisson process.

(= $x(t_1)-x(t_2)$) events in the time interval of length τ (= t_1-t_2) is given by

$$P\{m,\tau\} = \frac{(\lambda\tau)^m}{m!} e^{-\lambda\tau}, \qquad (2.3.7)$$

where $\lambda\tau$ is the mean: $E\{x(t_1)-x(t_2)\} = \lambda(t_1-t_2) = \lambda\tau$.

For example, the number of telephone calls at a switch board is a Poisson process with the intervals between successive calls being independent increments and having a distribution given by (2.3.7). Fig.2.3.2 shows a typical Poisson process.

Random walk

Random walk is a discrete time process with magnitude randomly jumping by +1 or -1 (or by +L, -L, where L is a constant) at each periodic instant.

The position, x(t), at the n-th period

$$x(nT) = x_1 + x_2 + \dots + x_n,$$

where T = time interval, and $\{x_i, i = 1,...n)\}$ is a family of independent and identically distributed random variables assuming values +1 or -1. Thus the position at any discrete time is obtained by integrating the random variables. Hence

$$x(nT) - x(nT-T) = x_n,$$

or

$$x(nT) = x_n/(1-q^{-1}),$$

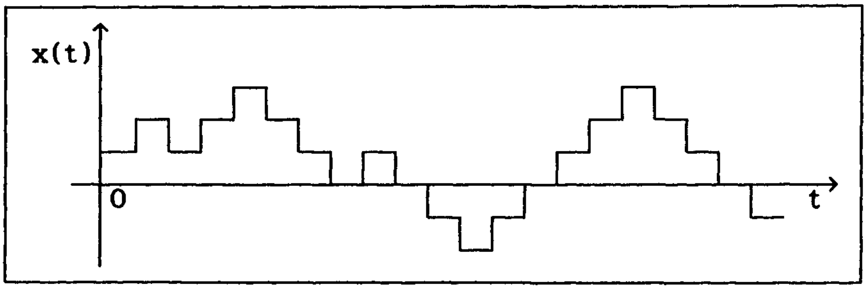


Figure 2.3.3 A typical random walk sequence.

where q^{-1} is a discrete time backward shift operator: $q^{-1}y(t) = y(t-T)$. It can be shown that

$$E\{x(i)\} = 0,$$

and

$$E\{x^2(nT)\} = nL^2.$$

Fig.2.3.3 shows an example of a random walk sequence.

2.4 TRANSFER-FUNCTION MODELS

Mainly three types of variables constitute a transfer-function model: y, the measured output of the process, u, the known (or measured) input to the process, and e, the noise or uncertainty in the model.

2.4.1 Some Basic Models

Autoregressive (AR) model

These models can be expressed as

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n) = e(k), \qquad (2.4.1a)$$

where $a_1,\dots,a_n$ are the model parameters; y(k) refers to the measurement of the output at time k. y(k-n), y(k-n+1), ..., y(k-1) are measurements of the output at successive time instants in the past; for example in the case of monthly data, these are data for successive months. e is referred to as the noise or the disturbance; it accounts for the errors in the measurements, the unaccounted for disturbances acting on the process and the modelling error.

The model (2.4.1a) can also be expressed as

$$A(q^{-1})y(k) = e(k), \qquad (2.4.1b)$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_nq^{-n},$$

q^{-1} being a unit discrete time backward shift operator.

Integrated autoregressive (IAR) model

This model is similar to the AR model except for the integrated noise structure

$$A(q^{-1})y(k) = \frac{e(k)}{\Delta}, \qquad \Delta = 1-q^{-1}. \qquad (2.4.2a)$$

So

$$A(q^{-1})\Delta y(k) = e(k),$$

that is

$$\Delta y(k) + a_1\Delta y(k-1) + \dots + a_n\Delta y(k-n) = e(k), \qquad (2.4.2b)$$

where $\Delta y(k) = y(k) - y(k-1)$.

Example

The yearly variation of the earth's rotation rate $\{y(k)\}$ shown in Fig.2.2.2 can be modelled as

$$\Delta y(k) = 1.0197\Delta y(k-1) - 0.6746\Delta y(k-3) + 0.508\Delta y(k-4)$$
$$-0.2361\Delta y(k-6) + 0.1981\Delta y(k-8) - 0.1487\Delta y(k-10) + e(k).$$

The model is parameterized using the method discussed in Secs.3.3.2–3.3.3. The data are given in Appendix 8B.

Autoregressive moving average (ARMA) models

Here the noise is represented by an extended sequence:

$$A(q^{-1})y(k) = C(q^{-1})e(k), \qquad (2.4.3a)$$

where

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_rq^{-1}.$$

Thus (2.4.3a) is given by

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n)$$
$$= e(k) + c_1e(k-1) + \dots + c_re(k-r). \qquad (2.4.3b)$$

Here, for estimation of the parameters, it will be necessary to know e(k-1), e(k-2),..., e(k-r) for which estimated values may be used. A typical approach for the estimation of the noise follows.

Following (2.4.3a),

$$y(k) = (1-A(q^{-1}))y(k) + C(q^{-1})e(k).$$

So, the estimate $\hat{y}(k|k-1)$ can be expressed as

$$\hat{y}(k|k-1) = -\hat{a}_1y(k-1) - \dots - \hat{a}_ny(k-n-1)$$
$$+ \hat{c}_1e(k-1) + \dots + \hat{c}_re(k-r), \qquad (2.4.4)$$

where $\hat{a}_1$, $\hat{c}_1$ etc. are estimated parameters. The estimate of e(k) is given by $\hat{e}(k) = y(k)-\hat{y}(k|k-1)$. Initially the values of e may be assumed to be zero.

Autoregressive integrated moving average (ARIMA) model

Similar to (2.4.2a), here an integrated noise structure is considered:

$$A(q^{-1})y(k) = \frac{C(q^{-1})e(k)}{\Delta}. \qquad (2.4.5)$$

That is

$$\Delta y(k) + a_1\Delta y(k-1) + \dots + a_n\Delta y(k-n)$$
$$= e(k) + c_1e(k-1) + \dots + c_re(k-r).$$

Autoregressive moving average model with exogeneous input (ARMAX)

This model is similar to ARMA, with additional input variable(s) incorporated:

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k), \qquad (2.4.6)$$

where

$$B(q^{-1}) = b_o + b_1q^{-1} + \dots + b_mq^{-m};$$

d is the time delay between the input u and the output y, that is a change in u results in a change in the output y after d time-steps. ARMAX and CARMA or Controlled ARMA models are of the same category.

Autoregressive integrated moving average model with exogeneous input (ARIMAX)

This model is similar to ARIMA model with additional exogeneous input variable(s) incorporated:

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + \frac{C(q^{-1})e(k)}{\Delta} ; \qquad (2.4.7)$$

The CARIMA (i.e. Controlled ARIMA) model has the same structure; the use of CARIMA models in process control features in Chapters 12 and 13.

Remarks
(1) The models discussed here are algebraically similar to the regression model

$$y(k) = a_o + a_1x_1(k) + \ldots + a_nx_n(k) + e(k).$$

Here, each regressor vector is a time series by itself.
(2) The estimation of autoregressive parameters can be more difficult than the moving average parameters due to the inherent coupling between the output y(k) and the corresponding time delayed variables y(k-1), y(k-2)· etc.

2.4.2 Model Structures

The representative estimation of the parameters and the validity of the model are the main considerations which influence the choice of the structure of the model. It is desirable that the estimates are true, yet the estimation procedure is simple; for this, one of the common requirements is that the noise should be uncorrelated with the data. Again, the long term validity of the model requires the stationarity of the data.

Preprocessing of data

Appropriate preprocessing of the data can help the estimation procedure and ensure validity of the estimates. The most common approach for preprocessing is differencing. The differencing for {y(k)} can be as follows:

(a) unit time-step differencing, e.g., $z(k) = y(k)-y(k-1)$,
(b) successive differencing, e.g., $x(k) = z(k)-z(k-i)$, $i \ge 1$.
(c) periodic differencing, e.g., $z(k) = y(k)-y(k-p)$, where p is the period length, etc.

Example
The German unemployment series $\{y(k)\}$ (Appendix 7E) shown in Fig.14.3.3 is modelled as

$$w(k) = -0.1164w(k-2) - 0.3039w(k-11) - 0.4062w(k-12) + e(k),$$

where $w(k) = z(k) - z(k-1)$, and $z(k) = y(k) - y(k-12)$. □□

Again, preprocessing of the data in terms of nonlinear transformation (discussed in Sec.8.2.3) can improve stationarity. Nonlinear transformation is also used to formulate linear-in-the-parameter models for nonlinear time series or processes.

Estimation and the noise process

The proper characterization of the noise is important from a parameter estimation point of view. The assumed noise structure has to conform to the requirements of the estimation method, as otherwise the estimates may not be valid. A convenient and hence popular assumption is that the noise process is white. In such cases the least squares method (Sec.3.3), which is one of the simplest methods of parameter estimation, can be used, as it produces optimal estimates.

In real life, the noise is rarely white with Gaussian distribution. Therefore the model is configured so that the noise has a convenient structure. The integrated noise structures considered in the case of IAR, ARIMA and ARIMAX models are produced with such an objective. The differencing of the data implicit with these models tends to result in the noise being uncorrelated with the data.

Consider the characteristics of the data from frequency content point of view. For every process, the frequency components present in the data are expected to lie within a certain range. Very low frequency components are responsible for the mean or zero frequency component (also referred to as the DC or average component, in analogy to electrical currents) and the trend component present in the data. Since the variations in the magnitude of these components are small, estimation of the corresponding parameters is difficult and hence should be avoided. Therefore, it is desirable to configure the model such that the low frequency components are not present; this can be implemented by high-pass filtering (Appendix 14A) of the data before estimation; simple time-differencing is a basic form of high-pass filtering.

In this connection, there are two important issues to be considered: (i) the noise associated with the data often contains high frequency components (typical examples are spikes, sudden rises or falls in the level, outliers etc.), and (ii) high-pass filtering tends to accentuate high frequency noise. The effect of such noise can be eliminated by low-pass filtering (Appendix 14A) of the data.

Thus modelling from the measured data has to satisfy three main requirements:

(i) the noise or the component of uncertainty in the model should be uncorrelated with the data,

(ii) it should not be required to estimate the parameters associated with low frequency components in the data,

(iii) the estimator should be protected from the effects of noise associated with the data.

These requirements can be satisfied as follows.

The integrated noise structure in an IAR, ARIMA or ARIMAX model, involves the differencing of the data, which amounts to high-pass filtering. One of the ways of implementing low-pass filtering at the same time is to use a noise observer polynomial ($T(q^{-1})$) as shown below. Consider an ARIMAX model (2.4.7):

$$\frac{A(q^{-1})y(k)}{T(q^{-1})} = \frac{B(q^{-1})u(k-d)}{T(q^{-1})} + \frac{C(q^{-1})e(k)}{\Delta T(q^{-1})}, \tag{2.4.8}$$

or

$$A(q^{-1})\Delta y_f(k) = B(q^{-1})\Delta u_f(k-d) + e(k), \tag{2.4.9}$$

where

$$y_f(k) = y(k)/T(q^{-1}), \; u_f(k) = u(k)/T(q^{-1}),$$

and it is assumed that $C(q^{-1}) = T(q^{-1})$. A typical choice for $T(q^{-1})$ is

$$T(q^{-1}) = \frac{(1-0.6q^{-1})}{0.4}.$$

Here, $T(q^{-1})$ acts as a first-order filter with a steady state gain of unity.

The differencing of the data together with the use of the noise observer $T(q^{-1})$ is akin to bandpass filtering (see Appendix 14A); so the data (Δy_f, Δu_f etc.) available to the estimator in (2.4.9) are effectively bandpass filtered data. It may also be noted that the model is so configured that the noise $\{e(k)\}$ works out to be uncorrelated by hypothesis.

Remarks

(1) If the frequency range of the noise components and that of the true signal overlap, the elimination of noise will also decimate the information in the data, which is undesirable. In such cases, the effects of noise may be reduced through optimal estimation discussed in Sec.6.6.

(2) The noise observer polynomial $T(q^{-1})$ may also be used in ARMA models (2.4.3) and ARMAX models (2.4.6).

(3) Models like (2.4.8) (i.e. incorporating the use of $T(q^{-1})$) feature in Chapters 5 and 12.

2.4.3 Other Models

Some transfer-function models which are popularly used in long range predictive control (see Secs. 12.4 and 12.5) are the pulse response and the step response models, which are discussed here.

Impulse response models

Consider an input signal $\{u(t)\}$ which is a pulse of

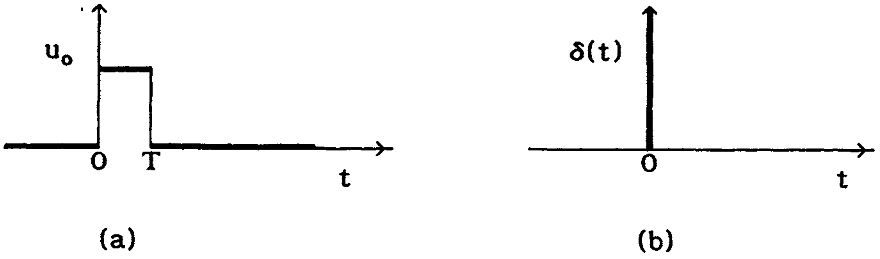


Figure 2.4.1 (a) A unit pulse, (b) a unit impulse.

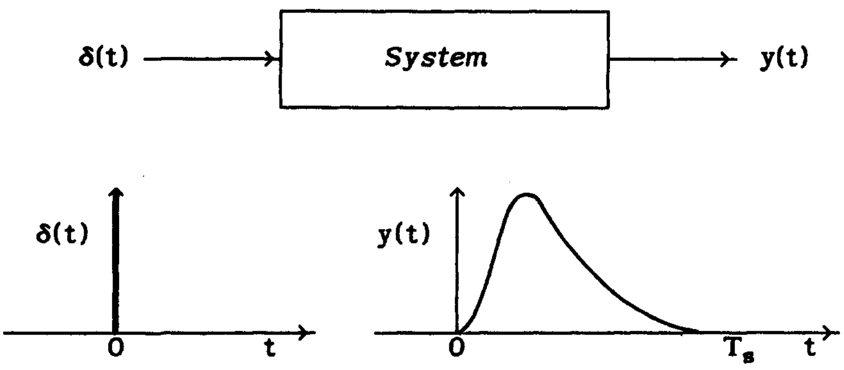


Figure 2.4.2 System response to an impulse input

magnitude u_0 and duration T as shown in Fig.2.4.1(a). The intensity of the signal is given by the area u_0T. As T is reduced, u_0 has to be increased if the intensity is to remain unchanged. This leads to the limiting condition with T tending to 0, when the pulse is of infinitesimal duration, and is of infinite magnitude; such a signal is called an *impulse* (Fig 2.4.1(b)). When the area under the impulse is unity, it is referred to as a *unit impulse*.

The unit impulse function, also known as the *Dirac delta function*, was originally defined by mathematician and physicist P.A.M. Dirac as

$$\int_{-\infty}^{\infty} \delta(t)dt = 1.$$

$$\delta(t) = 0, \qquad t \neq 0;$$

that is the unit impulse function δ(t) is zero except at t=0, where it covers a unit area.

The *impulse response* h(t), i.e. the response of a system with the unit impulse signal δ(t) as the input, completely characterizes a system in continuous-time representation:

$$y(t) = \int_{0}^{\infty} h(\tau)\delta(t-\tau)d\tau. \qquad (2.4.10)$$

For unit impulse input δ(t), h(t) = y(t).

In practice, when a system is excited by an input $\delta(t)$ at $t=0$, the output $y(t)$ will settle after a time T_s (see Fig.2.4.2), called the *settling time.* Hence (2.4.10) can be expressed as

$$y(t) = \int_0^{T_s} h(\tau)\delta(t-\tau)d\tau. \qquad (2.4.11)$$

The inputs that occurred earlier than the time T_s in the past do not influence the present output.

The impulse response is also referred to as the *weighting function.*

Remarks

(a) The convolution of two functions $f_1(t)$ and $f_2(t)$ is known as the *convolution integral*:

$$\begin{aligned} y(t) &= f_1(t)*f_2(t) \\ &= \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau. \end{aligned}$$

Thus Equation (2.4.11) is a convolution integral subject to the causality condition: $h(\tau) = 0$, for $\tau<0$.

(b) Equation (2.4.10) shows that the impulse response is directly given by the parameters of the continuous-time model.

Pulse response model

A unit pulse is defined as a pulse with unit magnitude and of unit sample-time duration:

$$\begin{aligned} &u(0) = 1, \\ &u(k) = 0, \qquad \text{for } k \neq 0. \end{aligned}$$

In discrete time representation, a single-input single-output system is completely characterized by the *pulse response* (also loosely called the impulse response), given by the response of the system to a unit pulse (Fig.2.4.3).

The discrete time equivalent of the convolution integral (2.4.10) is given by

$$y(k) = \sum_{j=o}^{\infty} h(j)u(k-j). \qquad (2.4.12)$$

For unit pulse input, $y(k) = h(k)$. Following (2.4.12),

$$\begin{aligned} y(k) &= h(0)u(k) + h(1)u(k-1) + h(2)u(k-2) + \dots \\ &\qquad + h(n)u(k-n) + \dots \end{aligned}$$

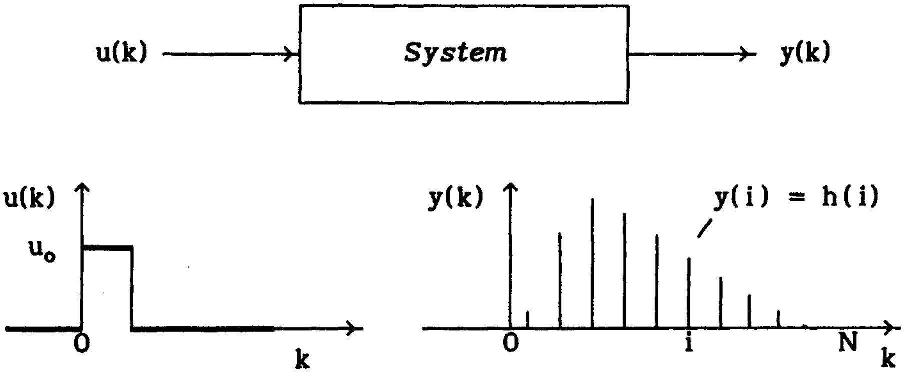


Figure 2.4.3 Pulse response of a system

That is

$$y(k) = \left(\sum_{j=0}^{\infty} h(j)q^{-j}\right)u(k)$$

$$= H(q^{-1})u(k), \qquad (2.4.13)$$

where $H(q^{-1})$ is referred to as *pulse-transfer function.* Since by definition, $u(0) = 1$, and $u(k) = 0$ for $k \neq 0$, the values of $y(k)$ for $k = 0, 1, 2$, etc. will be given by the coefficients of $H(q^{-1})$:

$y(0) = h(0)$,
$y(1) = h(1)$,
$y(2) = h(2)$, etc.

Usually $y(0) = 0 = h(0)$. If the settling time is N samples,

$$y(N+i) = h(N+i) = 0, \qquad i = 1,2,\ldots, \text{ etc.}$$

Thus the function h(k), called the pulse response, is obtained by applying a unit pulse input and measuring the system output at successive sampling instants.

If the system is expressed by the discrete time model

$$A(q^{-1})y(k) = B(q^{-1})u(k), \qquad (2.4.14)$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_nq^{-n},$$

$$B(q^{-1}) = b_o + b_1q^{-1} + b_2q^{-2} + \ldots + b_nq^{-n}.$$

y(k) can be expressed by the transfer-function model

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})} u(k),$$

$$= H(q^{-1})u(k),$$

which shows the equivalence between the transfer-function model and the pulse response model.

Remarks

(a) The transfer-function model (2.4.14) requires less parameters to describe the system behaviour compared to the pulse response model (2.4.13).

(b) The pulse response is directly given by the *parameter values* of $H(q^{-1})$ in the model: $y(k) = H(q^{-1})u(k)$.

(c) A process expressed by a pulse response model is open-loop stable, as otherwise the pulse response cannot be valid.

(d) If the process has a dead time d (i.e. if y does not show any change for d number of pulse durations after u is changed), $h_1 = h_2 = \ldots = h_{d-1} = 0$.

Step response model

A unit step u is defined as follows:

(a) in continuous-time representation,

$u(t) = 0$ for $t < 0$,

$u(t) = 1$ for $t \geq 0$,

(b) in discrete time representation,

$u(k) = 0$ for $k < 0$,

$u(k) = 1$ for $k \geq 0$.

Thus unit step is the first time-integral of a unit impulse (or a unit pulse) occurring at the time instant (or the sampling instant) at which the discontinuity appears in the step function. In discrete time representation, a step is equivalent to a train of pulses appearing at consecutive sampling instants. Hence

$$s_o = h_o,$$
$$s_1 = h_1 + h_o,$$
$$s_2 = h_2 + h_1 + h_o,$$
$$\vdots$$
$$s_p = \sum_{j=o}^{p} h_j \qquad (2.4.15)$$

etc.,

where $\{s_i\}$ and $\{h_i\}$ stand for step response and pulse

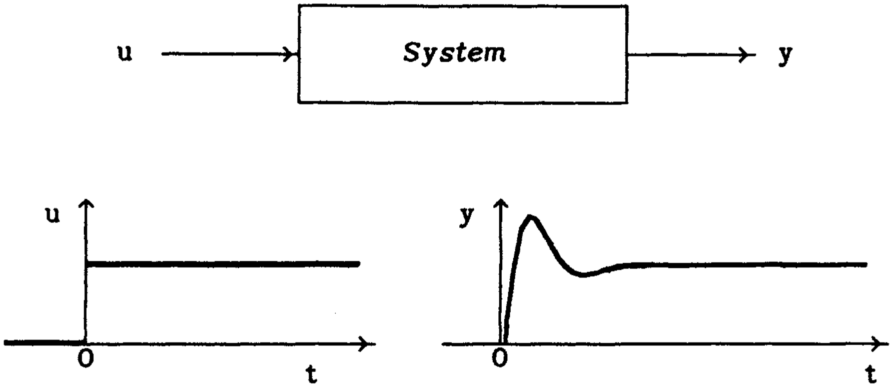


Figure 2.4.4 Step response of a second order system

response respectively. Thus the step response $\{s_i\}$ never converges. Following (2.4.13), the step response is given by

$$y(k) = \sum_{i=o}^{\infty} s_i u(k-i).$$

$$= S(q^{-1})u(k) = \frac{H(q^{-1})}{\Delta} u(k), \quad \Delta = 1-q^{-1}, \qquad (2.4.16)$$

where

$$S(q^{-1}) = \sum_{i=o}^{\infty} s_i q^{-1}$$

is the *step response transfer function* (see Fig.2.4.4).

2.5 MODELS BASED ON FREQUENCY DOMAIN ANALYSIS

A periodic or nonperiodic time series or signal can be composed of a number of sinusoidal components, each having a particular frequency. So the series can be modelled in sine and cosine trigonometric functions (or as exponential functions) of the constituent frequency components. In this connection, the sampling rate or the number of data points available within a certain time span is also important, because the sampling rate limits the highest frequency component that can be present in the sampled signal.

This section introduces the concepts of modelling the periodic and aperiodic signals in terms of the constituent frequency components. The frequency domain representation of a periodic signal in terms of a Fourier series and that of a nonperiodic signal in terms of a Fourier transform are described. Some of the most pioneering works in the area of frequency domain representations were done by the French mathematician J.B.J. Fourier (1768-1830).

The sampling aspects of a time series or signal are also studied in this section.

2.5.1 Representation of a Periodic Signal and the Fourier Series

Consider a continuous periodic signal f(t), with period T (Fig.2.5.1). By definition, a periodic signal repeats itself after the period T, that is

$$\begin{aligned} f(t) &= f(t+T) \\ &= f(t+nT), \qquad n = \pm1, \pm2... \end{aligned}$$

The objective is to represent f(t) in terms of functions of constituent frequency components. An obvious choice is to use sinusoidal functions. Define ω_0 to be the angular frequency corresponding to the time period T, i.e. $\omega_0 = 2\pi/T$. Since

$$\cos\omega t = \cos(\omega t\pm2\pi) \quad \text{and} \quad \sin\omega t = \sin(\omega t\pm2\pi),$$

$$\begin{aligned} \cos\omega_0 t &= \cos(\omega_0 t+2\pi n), \qquad n = \pm1, \pm2, \ldots \\ &= \cos\omega_0(t+nT), \qquad (2.5.1a) \end{aligned}$$

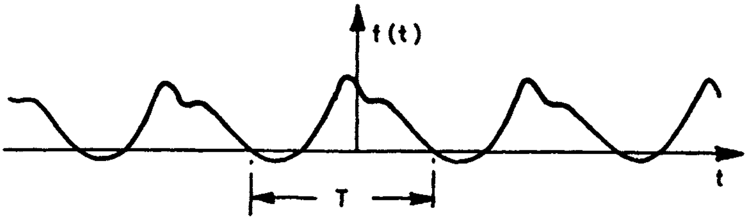


Figure 2.5.1 A continuous-time periodic signal f(t) having period length T.

Similarly,

$$\sin\omega_o t = \sin\omega_o(t+nT). \qquad (2.5.1b)$$

Hence it should be possible to represent f(t) in terms of the sinusoidal functions.

The Fourier series

Consider the infinite series

$$a_o + (a_1\cos\omega_o t+b_1\sin\omega_o t) + (a_2\cos 2\omega_o t + b_2\sin 2\omega_o t)$$
$$+ \ldots + (a_n\cos n\omega_o t+b_n\sin n\omega_o t) + \ldots$$

If this series converges to f(t), it will also converge to f(t+T), following (2.5.1), leading to the following deductions:

Deduction A: The summation of sinusoidal components with angular frequencies 0, ω_o, $2\omega_o$,..., $n\omega_o$, produces a periodic signal with period T, where $T = 2\pi/\omega_o$.

Here the zero frequency represents the constant component or the mean value. The converse of this result also holds as follows.

Deduction B: Any periodic signal f(t), with period T, can be decomposed into an infinite number of additive sinusoidal components with angular frequencies 0, ω_o, $2\omega_o$, ..., $n\omega_o$:

$$f(t) = a_o + \sum_{n=1}^{\infty}(a_n\cos n\omega_o t+b_n\sin n\omega_o t). \qquad (2.5.2)$$

The magnitudes of the coefficients a_i and b_i define the pattern of the function f(t). The series in (2.5.2) is called the *trigonometric Fourier series.*

One of the strong features of the sinusoidal functions in (2.5.2) is their property of orthogonality which is used in computing the coefficients a_n and b_n. The condition of orthogonality is stated as follows.

A set of continuous functions

$$\{v_m(t)\} = \{v_o(t), v_1(t), \ldots\}$$

is orthogonal over the interval (t_o, t_o+T), where t_o is arbitrary, if

$$\int_{t_o}^{t_o+T} v_m(t)v_n(t)dt = \begin{cases} D, \text{ if } m = n, & \text{D being a constant,} \\ 0, \text{ if } m \neq n. & \end{cases}$$

The sine and cosine functions in (2.5.2) show orthogonality as follows:

$$\int_{t_o}^{t_o+T} \cos m\omega_o t \,\cos n\omega_o t \, dt = \begin{cases} T/2, & m = n, \\ 0, & m \neq n. \end{cases} \quad (2.5.3a)$$

$$\int_{t_o}^{t_o+T} \cos m\omega_o t \,\sin n\omega_o t \, dt = 0, \text{ for all } m,n. \quad (2.5.3b)$$

$$\int_{t_o}^{t_o+T} \sin m\omega_o t \,\sin n\omega_o t \, dt = \begin{cases} T/2, & m = n, \\ 0, & m \neq n. \end{cases} \quad (2.5.3c)$$

Following (2.5.2) and (2.5.3), the coefficients a_i and b_i can be determined as follows.

$$a_o = \frac{1}{T}\int_{t_o}^{t_o+T} f(t)dt, \quad (2.5.4a)$$

$$a_n = \frac{2}{T}\int_{t_o}^{t_o+T} f(t)\cos n\omega_o t \, dt, \quad (2.5.4b)$$

$$b_n = \frac{2}{T}\int_{t_o}^{t_o+T} f(t)\sin n\omega_o t \, dt. \quad (2.5.4c)$$

The a_i and b_i coefficients are guaranteed to exist subject to the *Dirichlet conditions*:

(a) the periodic integral of $|f(t)|$ should exist, that is

$$\int_{t_o}^{t_o+T} |f(t)|dt < \infty, \quad \text{and}$$

(b) f(t) must be finite or have a finite number of discontinuities in one period.

Cosine series representation

The series (2.5.2) can also be expressed in terms of a cosine series, as

$$f(t) = c_o + \sum_{n=1}^{\infty} c_n \cos(n\omega_o t + \phi_n), \tag{2.5.5}$$

where

$$c_o = a_o, \quad c_n^2 = a_n^2 + b_n^2, \quad \phi_n = -\tan^{-1}\left(\frac{b_n}{a_n}\right).$$

c_o is the constant component, expressed as the average value of f(t) over one period as in (2.5.4a).

Exponential Fourier series

Instead of the sinusoidal functions, the periodic signal f(t) can also be expressed in terms of exponential functions (see Appendix 2):

$$f(t) = \sum_{n=-\infty}^{\infty} g_n e^{in\omega_o t}, \qquad i = \sqrt{-1}. \tag{2.5.6}$$

Equation (2.5.6) is the complex or exponential Fourier series representation of f(t). The coefficients g_n are given by

$$g_n = \frac{1}{T}\int_{t_o}^{t_o+T} f(t)e^{-in\omega_o t}dt, \tag{2.5.7}$$

where t_o is arbitrary, and $g_o = a_o$, as in (2.5.4a).

Thus the exponential Fourier series represents the spectrum of f(t) (that is the amplitudes of f(t) over various discrete frequencies), which is referred to as a *discrete spectrum* or a *line spectrum.*

2.5.2 Representation of a Nonperiodic Signal and the Fourier Transform

A nonperiodic signal can be expressed as a continuous sum of exponential functions of frequencies lying in the interval $-\infty < \omega < \infty$. The mathematical expression is developed as follows:

(a) The nonperiodic signal f(t) of finite length in time, τ, is considered to be a part of an augmented periodic signal, $f_a(t)$ with period, $T \geq \tau$ (Figs.2.5.2 and 2.5.3).

(b) The augmented periodic signal is expressed by the exponential series (2.5.6).

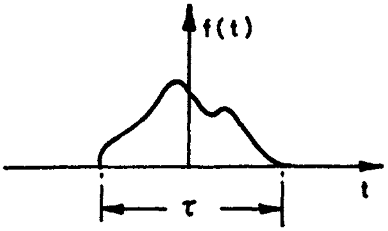


Figure 2.5.2 A nonperiodic process of length τ.

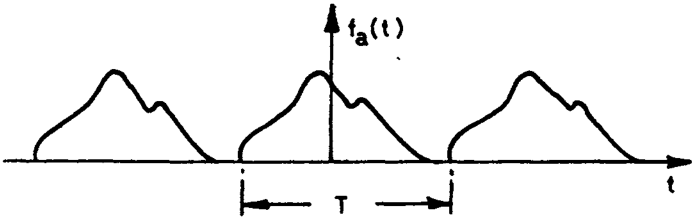


Figure 2.5.3 An augmented periodic process with period $T \geq \tau$.

(c) The limit conditions, $T \longrightarrow \infty$ and corresponding angular frequency, $\omega_o \longrightarrow 0$, applied to $f_a(t)$ leads to the continuous frequency representation of f(t).

Following (2.5.6-2.5.7) the periodic process $f_a(t)$ can be expressed by the exponential series

$$f_a(t) = \sum_{n=-\infty}^{\infty} g_{an} e^{in\omega_o t}, \qquad \omega_o = 2\pi/T$$

with the complex coefficients

$$g_{an} = \frac{1}{T}\int_{-T/2}^{T/2} f_a(t) e^{-in\omega_o t} dt.$$

As the time period T increases and tends to infinity, ω_o tends to infinitesimally small value or $d\omega$, and $n\omega_o$ becomes the continous angular frequency ω. Hence introduce

$$F(\omega) = \lim_{T \to \infty} T g_{an} = \lim_{T \to \infty} \int_{-T/2}^{T/2} f(t) e^{-in\omega_o t} dt. \tag{2.5.8}$$

Again

$$f(t) = \lim_{T\to\infty} f_a(t) = \lim_{T\to\infty} \sum_{n=-\infty}^{\infty} \frac{1}{T} F(\omega) e^{in\omega_0 t}$$

Since

$$\lim_{T\to\infty} \frac{1}{T} = \lim_{\omega_0\to 0} \frac{\omega_0}{2\pi},$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega, \tag{2.5.9}$$

where following (2.5.8),

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt. \tag{2.5.10}$$

The existence of $F(\omega)$ is subject to the condition that $f(t)$ is absolutely integrable in the time interval $(-\infty, \infty)$.

Equations (2.5.10) and (2.5.9) are referred to as the *Fourier transform* of $f(t)$ and the *inverse Fourier transform* of $F(\omega)$ respectively. Since $F(\omega)$ represents the frequency spectrum of $f(t)$, it is called the *spectral density function*.

Some common functions in time and their Fourier transform are shown in Fig.2.5.4; the following features may be noted:

(a) The Fourier transform of the rectangular pulse in Fig.2.5.4(a) is given by

$$F(\omega) = \tau \frac{\sin(\omega\tau/2)}{(\omega\tau/2)}.$$

This function is referred to as the *sampling function*.

(b) For the sine function in Fig.2.5.4(c) the phase of $F(\omega)$ at ω_0 and $-\omega_0$ will be $\pi/2$ and $-\pi/2$ respectively.

(c) It is evident from Fig.2.5.4(b) and Fig.2.5.4(c) that sinusoidal functions (and hence exponential functions) cause a frequency translation in the transform domain.

Remarks

(1) Both the Fourier transform and the Fourier series concern decomposition of a signal into constituent components having specific frequencies. The Fourier transform of an aperiodic signal is a linear combination of the

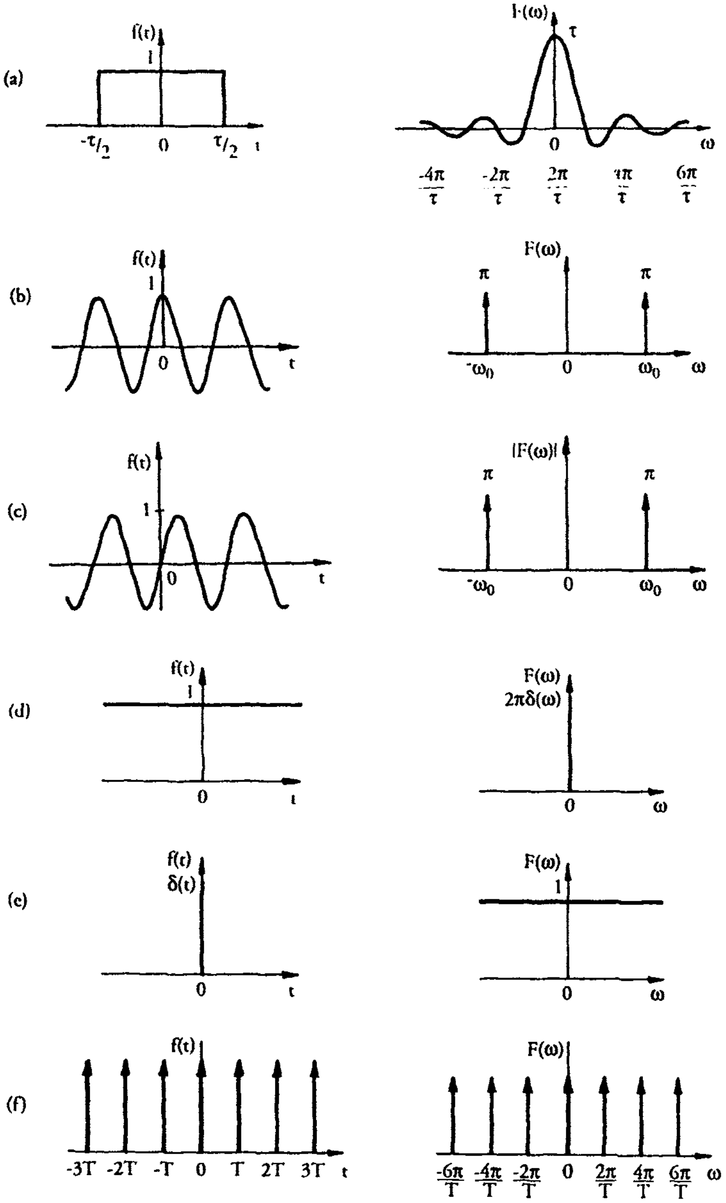


Figure 2.5.4 Some common time functions and their Fourier transform representations.
(a) A rectangular pulse, (b) a cosine function,
(c) a sine function, (d) a constant value,
(e) a unit impulse, (f) a uniform pulse sequence.

exponential functions of frequencies occurring at a continuum of frequencies (which means the corresponding frequency spectrum is a continuous spectrum), whereas the Fourier series of a periodic signal is a linear combination of the exponential functions of frequencies occurring at discrete frequencies (i.e. the corresponding frequency spectrum is a discrete or line spectrum).

(2) For a signal spanning a finite time τ, the values of the Fourier transform $F(\omega)$ at the points, $\omega = 2\pi n/\tau$, will be the same as the discrete spectrum produced by the Fourier series of the corresponding augmented periodic function with period τ. The envelope of the discrete spectrum due to the Fourier series will be the same as the envelope of the continuous Fourier transform spectrum.

(3) The Fourier transform is also an orthogonal transform.

2.5.3 Discrete-time Signals and their Fourier Transform

Although most real-life processes are inherently continuous-time processes (for example, the population of a country or say the temperature of the molten hot metal tapped from a blast-furnace), for ease of computation in modelling or other applications, a discrete time representation of the process is considered. A discrete time representation implies measurement or sampling of the process variable at discrete time intervals. The sampling rate (i.e. the number of samples per unit time) is of fundamental significance for any discrete time representation.

The sampling theorem

A data sequence or signal may contain a number of sinusoidal components.

The *sampling theorem* states that the sampling frequency f_s in number of samples per second or Hertz (abbreviated as Hz) has to be at least twice the highest frequency component, f_m, present in the signal (i.e. $f_s \geq 2f_m$). In other words, if a continuous time signal is sampled at a frequency f_s, the sampled signal will contain all the frequency components of the original signal which are less or equal to $f_c = f_s/2$. The frequency f_c is called the *Nyquist critical frequency*, and f_s is called the *Nyquist rate of sampling*.

There are two basic implications of the sampling theorem:

(i) Frequency components lower than or equal to f_c can only be used to form the original signal.

(ii) If the signal is not band limited to less than the Nyquist critical frequency, f_c, all the power spectral density outside the frequency range, $-f_c<f<f_c$, spuriously moves into the frequency range $(-f_c,f_c)$. This phenomenon is called *aliasing*.

The sampling theorem is illustrated with the help of an example (Fig.2.5.5). Consider a continuous-time signal f(t), which is band limited in frequency, i.e. the Fourier transform (Fig.2.5.5(b)) shows no frequency components beyond ω_m radians per second (or, f_m Hz, $\omega_m = 2\pi f_m$). Sampling of f(t) every T_s seconds is equivalent to considering convolution of f(t) with a uniform train of impulses $\delta_a(t)$ having the period T_s. The impulse train is referred to as the sampling function with the sampling period T_s and the fundamental angular frequency ω_s $(= 2\pi/T_s)$, as the sampling frequency. The convolved or sampled output f′(t) is a train of impulses having the magnitude of f(t) at equispaced time intervals T_s given by

$$f'(t) = f(t)\delta_a(t).$$

The Fourier transform of f′(t) (Fig.2.5.5g) shows clear separation of the frequency components when the sampling frequency $\omega_s>2\omega_m$.

The importance of the sampling frequency being at least twice the maximum frequency component present in the signal is illustrated in Fig.2.5.6. The spectrum of the sampled signal, $F'(\omega)$ fails to contain the true representation of $F(\omega)$ when $\omega_s<2\omega_m$. This phenomenon is called *aliasing*, when the higher frequencies in the signal appear as lower frequencies in the spectrum causing corruption of f′(t). Since the higher frequency components over the range $(\omega_m-\omega_s)$ are as it were folded back at the frequency $\omega_s/2$ to lower frequency range, this phenomenon of aliasing is also called *frequency folding*. To avoid aliasing, often an *anti-aliasing* filter is used which is basically a low-pass filter, attenuating frequency components higher than $\omega_s/2$; low-pass filtering is discussed in Appendix 14A.

The original signal f(t) can be recovered from the sampled signal f′(t) by passing the sampled signal through

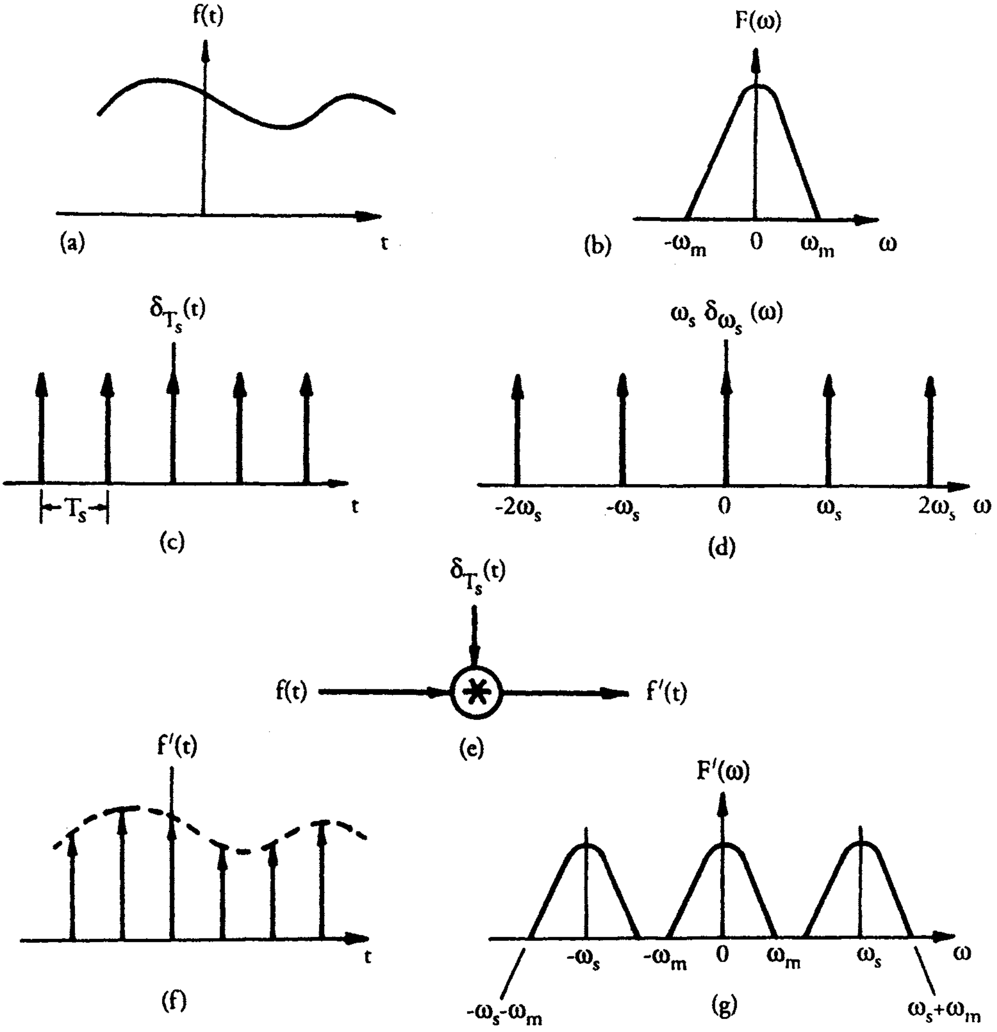


Figure 2.5.5 Sampling of a continuous-time signal and the frequency spectrum of the sampled signal.
(a) The continuous-time signal f(t),
(b) The spectrum of f(t).
(c) Uniform pulse train $\delta_{T_s}(t)$.
(d) Fourier transform of $\delta_{T_s}(t)$; sampling frequency = ω_s.
(e) Convolution of two signals f(t) and $\delta_{T_s}(t)$.
(f) The convolved signal f′(t) or the sampled signal f(t).
(g) $F'(\omega)$, the spectrum of f′(t), with $\omega_s > 2\omega_m$.

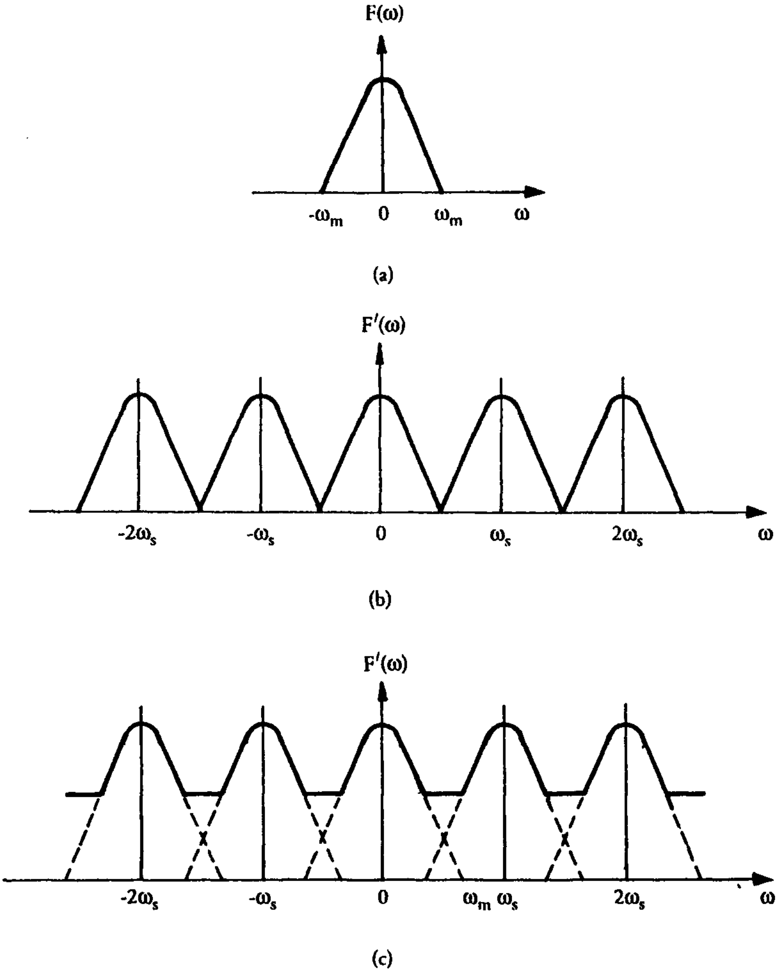


Figure 2.5.6 Effects of sampling frequency on the frequency spectrum of the sampled signal,
(a) spectrum of original signal,
(b) spectrum of sampled signal with $\omega_s = 2\omega_m$,
(c) spectrum of sampled signal with $\omega_s<2\omega_m$; aliasing resulting from low sampling rate.

an ideal low-pass filter having a cut-off frequency greater than ω_m but less than $(\omega_s-\omega_m)$. A separation between ω_m and $(\omega_s-\omega_m)$ is preferred; in practice, most designers like to have a sampling rate at least 20% higher than the minimum desirable rate based on frequency considerations.

Discrete Fourier transform and fast Fourier transform

The discrete Fourier transform (DFT) is a way of specifying the sinusoidal signals that constitute a finite discrete data sequence. Given any N-long data sequence {x(j)} = {x(0), x(1), ..., x(N-1)}, its DFT is defined as

$$x_w(m) = \frac{1}{N}\sum_{j=o}^{N-1} x(j)\exp(-i(2\pi/N)mj), \qquad m = 0,1,\ldots,N-1, \tag{2.5.11}$$

and the inverse DFT is defined as

$$x(j) = \frac{1}{N}\sum_{m=o}^{N-1} x_w(m)\exp(i(2\pi/N)jm), \qquad j=0,1,\ldots,N-1. \tag{2.5.12a}$$

If the data are complex, the DFT $x_w(m)$ will be the complex valued density functions of the N sinusoidal components, having freqencies 0, 1/N, 2/N,..., (N-1)/N; the frequencies are the normalized frequencies. If the data are real valued (which is usually the case in the present context), the frequencies of the constituent sinewaves are 0, 1/N, 2/N, ..., (N/2)/N; if N is odd, the highest frequency component is (N-1)/2N. The finest resolution between two adjacent components produced by DFT is 1/N.

For the real valued data sequence, {x(j)} can also be expressed as

$$x(j) = x(0) + \sum_{n=1}^{N/2}(a_n\cos((n/N)2\pi j) + b_n\sin((n/N)2\pi j)), \tag{2.5.12b}$$

where x(0) is the average value of the sequence {x(j)}, N is the length of the sequence (assumed to be even), and

$$a_n = \frac{2}{N}\sum_{j=1}^{N} x(j)\cos((n/N)2\pi j),$$

$$b_n = \frac{2}{N}\sum_{j=1}^{N} x(j)\sin((n/N)2\pi j).$$

There are various efficient algorithms for the computation of the DFT; the most well known of these are collectively referred to as the *fast Fourier transform* (FFT) algorithms.

Example 2.5.3 Model the yearly variations in the Earth's rate of rotation (Appendix 8B) using Fourier transform

The concerned series (see Fig.2.2.2) has 150 data points; it is padded with zeros to 1024 points, and a 1024-point FFT is

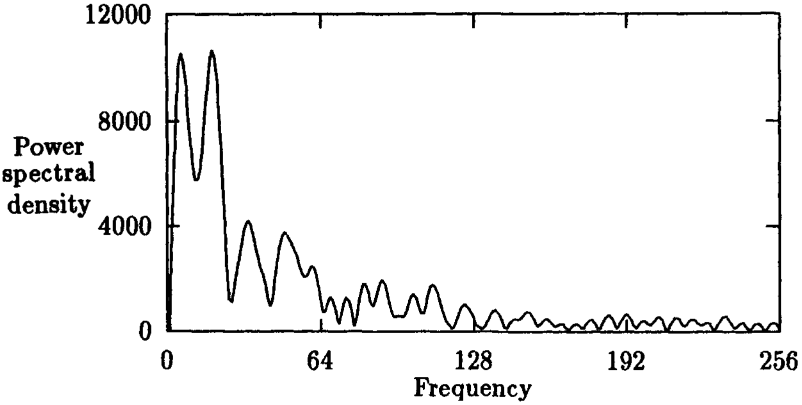


Figure 2.5.7 The frequency spectrum of the series on the yearly variations of the earth's rate of rotation.

performed on the augmented series. The frequency spectrum obtained is shown in Fig.2.5.7. The mean or average component and the 9 strongest sinusoidal components in order of magnitude are selected which have the normalized frequencies f_i (i=1 to 9) given by

19/1024, 6/1024, 20/1024, 18/1024, 7/1024, 5/1024,
21/1024, 17/1024, 8/1024.

The series may be modelled as

$$y(k) = a_0 + \sum_{i=1}^{9} b_i \sin(2\pi k f_i) + e(k),$$

where e is the noise, and k = 1,...,150. The least squares estimation is used to estimate the parameters, which are obtained as follows.

a_0 = -200.15, b_1 = 11653.86, b_2 = -5716.47, b_3 = -6891.67,
b_4 = -9697.50, b_5 = 4644.21, b_6 = 2758.74, b_7 = 1611.39,
b_8 = 3458.48, b_9 = -1431.74.

Remarks

(1) Here, although 150 data points were available, a 1024-point FFT was performed for higher resolution in the detection of the frequency components present, which is 1/1024 here. The concerned extension of the data series by zeros does not affect the result.

(2) In the model, only sine terms are considered for the sake of simplicity; cosine terms could also be considered.

2.5.4 Modelling of a Periodic Signal

There are two basic features of a periodic signal, from the sampling point of view:

(i) the smallest length of time for which the data are available, and

(ii) the period-length or time period at which the pattern tends to repeat.

Consider the modelling of any nearly periodic series with monthly data and yearly periodicity. Here,

(i) the sampling period, T_s is one month, i.e. $T_s = 1$, and

(ii) the period tends to repeat every 12 months, i.e. the time period, $T = 12T_s$.

Any series which has a periodicity of T, will have the fundamental frequency component, $\omega_o = 2\pi/T$, and can be modelled as a linear combination of cosine and sine functions of ω_o and its harmonics $2\omega_o$, $3\omega_o, \ldots$, and an average (or constant) term.

Again, according to the sampling theorem (Sec.2.5.2), if a signal or time series is sampled at T_s intervals, that is $\omega_s = 2\pi/T_s$, the maximum frequency component that can be present in the sampled data is given by $\omega_m \le \omega_s/2$.

Example 2.5.4 Model the monthly Indian rainfall series (Appendix 7F).

This series (shown in Fig.2.2.1) is a monthly data series, and has yearly periodicity.

For modelling, assume $T_s = 1$, $\omega_s = 2\pi$. So

(a) $\omega_m \le 6\omega_s/12$, since $T = 12$, $\omega_o = 2\pi/12$,

(b) the frequency components contained in the sampled signal are

$$2\pi(1/12),\ 2\pi(2/12),\ 2\pi(3/12),\ 2\pi(4/12),\ 2\pi(5/12) \text{ and } 2\pi(6/12).$$

Using the trigonometric Fourier series (2.5.12b), for the d-th year,

$$f_d(t)=a_o+\sum_{n=1}^{6}(a_n\cos((n/12)2\pi t) \quad + \quad b_n\sin((n/12)2\pi t)), \tag{2.5.13}$$

where $t = 1,2,\ldots,12$, and the values of the time series $f_d(t)$ are known. For $n = 6$,

$$\cos((n/12)2\pi t) = \cos\pi t = (-1)^t, \text{ and}$$

$$\sin((n/12)2\pi t) = \sin\pi t = 0$$

Again,

$$a_n = \frac{2}{12}\sum_{t=1}^{12} f_d(t)\cos((n/12)2\pi t), \text{ and}$$

$$b_n = \frac{2}{12}\sum_{t=1}^{12} f_d(t)\sin((n/12)2\pi t),$$

where n = 1,2,...,5, and

$$a_o = \frac{1}{12}\sum_{t=1}^{12} f_d(t), \qquad a_6 = \frac{2}{12}\sum_{t=1}^{12} f_d(t)(-1)^t.$$

The values of the coefficients a_n and b_n so obtained pertain to the data for the d-th year.

For the present series for the year 1941,

a_o=54.28, a_1=-57.41, a_2=1.22, a_3=10.06, a_4=-5.58, a_5= 3.15, a_6=-4.01,

b_1=-57.13, b_2=56.15, b_3=-16.58, b_4=8.64, b_5=-4.82, b_6=0,

in the model (2.5.13).

Remarks

(1) Since data for a number of years are available the coefficients a_n and b_n in (2.5.13) may also be estimated using the method of least squares; for example in the present case, there are 12 sets of data available for each year for the estimation of the 12 parameters a_o to a_6 and b_1 to b_5.

(2) Unlike the present case, if N is odd, there will be (N-1)/2 sinusoidal components in the data $f_d(t)$ having frequencies $n\omega_o t$, where

$$n = 1,\ 2,\ \ldots,\ \frac{N-1}{2},$$

in addition to an average or the zero frequency component.

2.6 STRUCTURAL MODELLING

In structural modelling, instead of modelling the composite series as it is, each of its structural components is separately modelled.

2.6.1 A Basic Model

A basic structural model of any series or signal $\{y(k)\}$ has three components, namely the trend (y_{tr}), the seasonal component (y_p), and the irregular component (e):

$$y(k) = y_{tr}(k) + y_p(k) + e(k). \qquad (2.6.1)$$

In terms of frequency descriptions, the trend is the slowly varying average or low frequency component, and the seasonal component contains all the frequency components that are responsible for seasonal variations. As such, the trend and the seasonal component together should suffice to describe a seasonal process. The uncertainties and unmodelled dynamics account for the irregular component. The model is called *structural* because its components convey structural information.

The trend component can be extracted by centred moving averaging (Appendix 4) or through bidirectional filtering (Sec.14.3.1) etc. Once the trend component is separated, the seasonal component in the remaining part of the signal may be modelled using any suitable method like the trigonometric functions approach (Sec.2.5.1), the Box and Jenkins approach (Sec.4.3) or the orthogonal transformation based methods (discussed in Chapters 7 and 11). Structural models can also be formulated using the state-space descriptions as shown in Sec.6.4.

Example: The German unemployment series (Appendix 7E), shows a trend component associated with a periodic component. In Example 14.3.1(2), the trend component is separated using bidirectional filtering (with α=0.9, in (14.3.1)). The structural components are shown in Fig.14.4.3.

2.6.2 Models with Multiple Periodic Components

The seasonality in the time series need not be explicitly due to a single periodic component for the structural modelling to be applicable. There can be multiple periodic components (with different period lengths) present in the series, which can be expressed as

$$y(k) = y_{tr}(k) + y_{p1}(k) + y_{p2}(k) +...+ y_{pN}(k) + e(k),$$

where each sequence $\{y_{pi}(k)\}$ (for $i = 1$ to N) is periodic with a repeating pattern; the magnitudes of the patterns of $\{y_{pi}(k)\}$ over different periodic segments may not be the same.

The trend component may be separated the same way as in Sec.2.6.1. The remaining part of the signal will constitute a combination of multiple repetitive pattern components and noise. The successive separation of the periodic components y_{pi} ($i = 1$ to N) can be performed using singular value decomposition as discussed in Sec.11.4; each of the periodic components as well as the trend may now be modelled individually as desired.

Typically a periodic component, y_{pi}, may be modelled as

$$\{y_{pi}(k)\} = \{u_{1(pi)}v_{1(pi)}^T\},$$

where $v_{1(pi)}^T$ represents the periodic pattern of the i-th periodic component, and the elements of $u_{1(pi)}$ are the scaling factors associated with the successive periods; the sequence of elements of $u_{1(pi)}$ may be modelled as a time series.

Example 11.4.5(1), studies a case where a noisy composite signal has been decomposed and three periodic components have been produced. The results are presented in Figs.11.4.2 and 11.4.3. The modelling of the individual periodic components is treated in Secs.7.7-7.8.

2.7 CONCLUDING REMARKS

The aim of this chapter has been to introduce the broad perspective of modelling.

The model design is based on the available data and other information on the process. So the representativeness of the model depends both on the representativeness of the data and the accuracy of the information available, and the way the available information is used in modelling. There is no *best design* of a model; one rather heuristically aims at the best possible compromise between the various contributing factors and hopes the model to behave like the actual process. Some of the important factors are the model structure, the uncertainty associated with the model, and the long-term validity of the model.

A real-life problem may not conform to the familiar model structures. The problem has to be appropriately configured (e.g., with suitable preprocessing of the data etc.), so that the imprecision or vagueness in the underlying assumptions is minimized. It is important that the modelling problem is so formulated that the estimation of the parameters with an acceptable degree of precision is feasible. In this connection, the characterization of the uncertainty or noise associated with the model deserves careful consideration. There is a common tendency to assume convenient idealistic characteristics of the residual component referred to as noise, to satisfy the preconditions of the validity of the estimation methods. It is important for the user that the assumptions used are known and their validity is ensured; in real life the validity of the idealistic assumptions about the nature of the data and the noise can be only approximately assured and hence the resulting estimates work out to be approximately true.

The numerical robustness of the estimation and other operations used for modelling also deserve careful consideration, as this concerns the validity of the model.

The model should preferably be as simple as possible. The ultimate objective is to produce a model which is representative and which continues to remain valid, with relevant adaptation if necessary.

REFERENCES

Remarks: There are many books on various aspects of process modelling, only some of which are listed here. Two authoritative texts on stochastic processes are [4,13]. A broad coverage of modelling methods appears in [11], and modelling, along with estimation aspects, is detailed in [8,2,5]. Books covering specific application areas are another class, e.g., [1,15]; state-space modelling has been studied in [3,16], GMDH is explored in [6]; many books are devoted to the area of neural network modelling, e.g., [14]. Models based on frequency domain analysis are detailed in many texts, for example [9,10,12]. Structural modelling features in [7]; see also Sec.11.5.

[1] Bartholomew, D. (1973): *Stochastic Models for Social Processes*, Wiley, New Work.

[2] Box, G.E.P., and G.M.Jenkins (1976): *Time series Analysis Forecasting and Control*, Holden-day, San Francisco, CA.

[3] Cadzow, J.A., and H.R. Martens (1970): *Discrete-time and Computer Control Systems*, Prentice-Hall, Inc., Englewood Chiffs, N.J.

[4] Doob, J.L. (1953): *Stochastic Processes*, Wiley, New York.

[5] Draper, N., and H. Smith. (1981): *Applied Regression Analysis*, 2nd edn., Wiley, New York.

[6] Farlow, S.J. (Ed.) (1984): *Self-organizing Methods in Modelling: GMDH Type Algorithms*, Marcel Dekker, New York.

[7] Harvey, A.C. (1990): *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge Univ. Press, Cambridge.

[8] Kashyap, R.L., and A.R. Rao (1976): *Dynamic Stochastic Modelling from Empirical Data*, Academic Press, New York.

[9] Kay, S.M. (1988): *Modern Spectral Estimation*, Prentice Hall Inc., Englewood Cliffs, New Jersey.

[10] Lathi, B.P. (1974): *Signals, Systems and Controls*, Harper and Row, New York.

[11] Murthy, D.N.P., N.W. Page, and E.Y. Rodin (1990): *Mathematical Modelling*, Pergamon Press, Oxford.

[12] Oppenheim, A.V., A.S. Willsky with I.T. Young (1983): *Signals and Systems*, Prentice-Hall, Englewood Cliffs.

[13] Papoulis, A. (1991): *Probability, Random Variables, and Stochastic Processes*, 3rd edn., McGraw-Hill, New York.

[14] Simpson, P.K. (1990): *Artificial Neural System*, Pergamon Press, New York.

[15] Soong, T.T. (1981): *Probabilistic Modelling and Analysis in Science and Engineering*, Wiley, New York.

[16] Strejc, V. (1981): *State Space Theory of Discrete Linear Control*, Wiley, Chichester.

[17] Wold, H.O.A. (1954): *A Study in the Analysis of Stationary Time Series*, 2nd edn., Almqvist and Wiksell, Uppsala.

CHAPTER 3

PARAMETER ESTIMATION

Correct parameterization and estimation ensures the representativeness of a model.

3.1 INTRODUCTION

System identification is a prerequisite to adaptive prediction and control; it concerns the generation (for example through specific experimentation) and collection of information, revealing the characteristic behaviour of the process, and development of a mathematical representation of the process. Thus while *parameter estimation* concerns the determination of the numerical values of the parameters of the process model which best describe the dynamics of the process, identification involves model structure selection, collection of relevant information, parameter estimation, and model validation. The nature of the model is very much process and problem dependent, as discussed in Chapter 2. This chapter is primarily devoted to the problem of parameter estimation, model order selection and validation.

There are different methods of parameter estimation. The suitability of a method depends on the quality of information contained in the data, the conceptual model structure and the application concerned. A detailed study of the estimation methods is beyond the scope of this book. The discussions are focused mainly on the least squares (LS) method, which is a basic method for parameter estimation. The quality of the estimates are shown to depend on the nature of the noise, and the richness of the information contained in the data. Both the off-line and the recursive implementations of the LS estimator are presented, and the computational aspects are studied. It has been shown that using orthogonal decomposition, ill-conditioned LS estimation problems can also be solved.

One constraint of the LS method is that the noise needs to be uncorrelated with the measurement of the dependent variable; there are many other estimation methods which can be used without any such restriction. This chapter includes discussions on three such methods, namely, the instrumental variable method, the maximum likelihood method, and the Koopmans-Levin method implemented using the singular value decomposition. The instrumental variable method avoids the requirement of the noise to be uncorrelated with the data, through the introduction of some auxiliary variables referred to as instrumental variables. The maximum likelihood method involves nonlinear optimization of the statistical information contained in the data; the method can produce parameter estimates with most of the desired properties, although at the cost of relatively intensive computation. The Koopmans-Levin method can handle noise associated with both the dependent as well as the independent variables, and can produce estimates equivalent to approximate maximum likelihood estimators.

Proper model structure selection and testing of the validity of a model are necessary for representative modelling. The study includes the Akaike Information Criterion (AIC), the methods of robust modelling using subset selection, and cross validation. The Akaike Information Criterion, which provides a statistical estimate of the appropriateness of a model, is a popular method for model order selection. Cross validation permits the test for the validation of a model through the use of a set of data, which has not been used for parameter estimation. Subset selection can be used for the selection of specific variables to form the most representative model. Two classes of subset selection have been studied: (a) subset selection from an information set, and (b) selection of independent variables in a regression problem. The implementation of subset selection through singular value decomposition and some special forms of QR factorization has been discussed. Subset selection is a powerful approach with enormous application prospects in the areas of identification, estimation and control; it has been applied in varied classes of problems in this book.

The linear regression and the least squares estimation methods are developed in Sec.3.2. The computational aspects discussed in Sec.3.3 concentrate on numerically robust and computationally efficient implementations; there is particular stress on LS estimation using orthogonal transformation.

Estimation with orthogonalized regressors is also presented. The recursive LS algorithm and its implementation are treated in Sec.3.4. This is followed by an exposure to the instrumental variable method, the maximum likelihood method, and the Koopmans-Levin method of estimation, in Sec.3.5. Finally AIC, model selection and cross validation are treated in Sec.3.6, wherein the problem of best subset AR modelling and the application of subset selection in regression problems are also discussed.

3.2 LINEAR REGRESSION AND THE LEAST SQUARES METHOD

Linear regression concerns linear-in-the-parameters representation, relating one or more independent variables with the dependent variable; least squares (LS) is a method for the estimation of the parameters of the linear regression. The term *independent* means 'independently appearing' in the regression equation; the independent variables may not be statistically independent. The present discussion on the LS estimation is confined to the off-line method only; the online method of recursive least squares estimation is discussed in Sec.3.4.

Linear regression

Consider the following models:

(a) $$y(k) + a_1y(k-1) +\ldots+ a_Ny(k-N) = b_1u(k-1) +\ldots+ b_Nu(k-N) + e(k), \tag{3.2.1}$$

(b) $$y(i) = \beta_1x_1(i) +\ldots+ \beta_px_p(i) + e(i), \tag{3.2.2}$$

In (3.2.1), $y(k)$ is the dependent or the response variable, and $y(j)$ and $u(j)$, for $j = (k-1)$ to $(k-N)$, are the variables on which y depends; $e(k)$ is the noise or the modelling error term. For input-output processes, y and u are the output and the input measurements respectively. In (3.2.2), the i-th observation of the dependent variable y is expressed in terms of the i-th observations of the variables, x_1 to x_p. Note that the models (3.2.1) and (3.2.2) have the same algebraic structure.

A concise vector expression for (3.2.1) is given by

$$y(k) = \mathbf{h}^T(k)\theta + e(k), \tag{3.2.3}$$

where **h**, the data vector, and θ, the parameter vector, are given by

$$\mathbf{h}(k) = [-y(k-1)\ -y(k-2)\ \ldots\ -y(k-N)\ u(k-1)\ \ldots\ u(k-N)]^T,$$

$$\theta \ = \ [a_1\ a_2\ \ldots\ a_N\ b_1\ b_2\ \ldots\ b_N]^T.$$

Let θ be assumed to be an n-vector. The models (3.2.1), (3.2.2) or (3.2.3) are called the *linear regression* models, and the data elements, that is the elements of **h**(k) in (3.2.3) are called the regressors. Following (3.2.3), if the measurements are available over the time (k-m+1) to k:

$$y(k-1) = \mathbf{h}^T(k-1)\theta + e(k-1),$$

$$y(k-2) = \mathbf{h}^T(k-2)\theta + e(k-2),$$

$$\vdots$$

$$y(k-m+1) = \mathbf{h}^T(k-m+1)\theta + e(k-m+1),$$

or in concise matrix notations

$$\mathbf{y} = \mathbf{H}\theta + \mathbf{e}, \tag{3.2.4}$$

where

$$\mathbf{y} = \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-m+1) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}^T(k) \\ \mathbf{h}^T(k-1) \\ \vdots \\ \mathbf{h}^T(k-m+1) \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e(k) \\ e(k-1) \\ \vdots \\ e(k-m+1) \end{bmatrix},$$

and the n parameters, θ, are based on all the data available from time k to (k-m+1). The model (3.2.4) is referred to as the *linear mutiple regression* model.

Remarks

(1) The term *linear* in linear regression implies that the expression is linear in the parameters θ, and in the error e, in (3.2.3). Nonlinearly transformed observations may be used as the data (i.e. in **h** in (3.2.3)), for example,

$$y(t) = \alpha_o + \alpha_1 t + \alpha_2 t^2 + \ldots + \alpha_n t^n,$$

where $\alpha_o, \ldots, \alpha_n$ are the parameters and 1, t, $t^2, \ldots, t^n$ are the regressors.

(2) e(k) in (3.2.3) may comprise the measurement error in y(k), external disturbances or the modelling error in $\mathbf{h}^T\theta$.

3.2.1 Formulation of LS Estimator

Given m sets of observations y(k) and H(k), the objective is to estimate the n parameters of θ in (3.2.4). The least squares estimation is based on the minimization of the scalar cost function,

$$J(\theta) = \frac{1}{2}\sum_{i=0}^{N-1}(y(k-i)-h^T(k-i)\theta)^2 \tag{3.2.5}$$

$$= \frac{1}{2}e^Te$$

$$= [y-H\theta]^T[y-H\theta].$$

The minimization of $J(\theta)$ with respect to θ is given by

$$\frac{\partial J(\theta)}{\partial\theta} = [y-H\theta]^T[-H] = 0. \tag{3.2.6}$$

Again

$$\frac{\partial^2 J(\theta)}{\partial^2\theta} = H^TH \geq 0,$$

since the matrix H^TH is positive semidefinite, and hence the function $J(\theta)$ has a minimum, given by (3.2.6). Hence the least squares estimate $\hat{\theta}$ can be obtained as

$$H^TH\hat{\theta} = H^Ty. \tag{3.2.7}$$

$$\hat{\theta} = [H^TH]^{-1}H^Ty. \tag{3.2.8}$$

Remarks

(a) Following (3.2.6),

$$[y-H\hat{\theta}]^TH = e^TH = 0,$$

which is the orthogonality condition that LS estimation must satisfy, that is the LS estimate $\hat{\theta}$ has to be so chosen that the consequent error vector **e** is orthogonal to each of the columns of **H**. The equation (3.2.7) is called the *normal equation*.

The LS estimation (3.2.8) is a linear transformation on **y**, and hence the LS estimator is referred to as a *linear* estimator.

(b) The LS estimator requires m>n, that is the number of measurements (m) to be greater than the number of parameters (n); usually m>>n. If **H** has a full column rank, or if m = n, or if the rank of **H** = min(m,n), the LS solution (3.2.8) is unique. If rank(**H**) < n, the estimated parameters will not be meaningful.

(c) Following (3.2.3), the term $\mathbf{h}^T(k-i)\hat{\theta}$ may be considered to be the estimation or prediction $\hat{y}(k-i)$. Hence, the error vector, $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$, may be considered to be the prediction error vector; so the LS method may be referred to as a *prediction error method.*

(d) Although **y** and **H** are stated as in (3.2.4), the data in **y** or **H** need not be ordered in any particular sequence.

3.2.2 Features and Properties

The quality of the LS estimates depends on

(i) the richness of information contained in H(k), and

(ii) the statistical properties of the noise sequence {e(k)}.

All the characteristic features are related to these two factors.

Richness and quality of information in the data

It is desirable that the data are rich in information, so that complete representation of the process dynamics is available to the estimator through the data; at the same time, the data should be balanced to the desired extent.

Rank deficiency in LS estimation

From an algebraic point of view, lack of information in the LS estimation shows up in the rank deficiency of **H**. In this connection the following aspects need some attention.

(a) multicollinearity between the different regressors,

(b) the data being too steady,

(c) some independent variables being nearly orthogonal to the dependent variable **y**.

Remark: Multicollinearity

In mathematics, *collinearity* means the property of several points being on the same line. The term *multicollinearity* or simply *collinearity* refers to one regressor or a set of regressors (collectively) being a linear function of a set of other regressors. Collinearity in **H** will show up in one column (or a set of columns) of **H** being a linear combination of other columns. Note that collinearity is a *phenomenon*. Collinearity or *near* collinearity has to be avoided for proper estimation of parameters.

□□

Algebraically, (a) will show up in redundancy in the columns of **H**, (b) will show up in redundancy between the rows of **H**, and both can result in **H** being singular and $\mathbf{H}^T\mathbf{H}$ being noninvertible; such classes of rank deficient LS estimation problems are discussed in Sec.3.3.3. It is desirable to eliminate the variables from the regression problem that causes rank redundancy in **H**.

It will be ideal, if the regressor vectors are mutually orthogonal; in such cases, the parameters associated with the individual regressors will remain unchanged even if some regressors are dropped.

The set of relatively steady data contains little information, and hence should be preferably eliminated in advance. For the same reason, the estimation of a constant term (i.e. with unity in the corresponding column in the data matrix **H**) is avoided; a model with integrating noise structure (for example an ARIMA model, as discussed in Sec.2.3.2) can absorb and exclude such constant terms. For a causal process it is desirable to have an input which is constantly changing (also referred to as 'persistently exciting'), and is expected to be able to excite all the modes of the process. If the data are rich, the information matrix $\mathbf{H}^T\mathbf{H}$ will be nonsingular.

Contrary to the conventional understanding, regressor variables which are orthogonal to **y** are *not* redundant, and should not be excluded from the estimator formulation. *The relationship within linear regression is a group phenomenon*; it is not possible to ascertain the relative importance of individual regressors through tests like correlation analysis etc. against the output variable. Subset selection in regression is discussed in Secs.3.6.2-3.6.4.

Uniformity of the data

The columns of **H** should be balanced, that is should contain energy of the same order. This can be ensured through normalization of the individual columns. The normalization of the columns is also necessary before subset selection.

The LS estimates can be seriously affected by outliers, that is abnormally large or small observations, which must be eliminated or truncated in advance.

Weighted least squares estimation

Differential importance may be ascribed to the data in **H** by appropriate weighting introduced in the cost function:

$$J_w(\theta) = \frac{1}{2}[y-H\theta]^T W[y-H\theta].$$

If $\mathbf{W} = \mathbf{I}$, $J_w(\theta) = J(\theta)$. If $\mathbf{W} = \text{diag}\ [w_1, w_2, w_3, \ldots, w_n\ ]$,

$$J_w(\theta) = \frac{1}{2} \sum_{i=1}^{N} w(i)e^2(k+i).$$

The elements of **W** are appropriately chosen to increase (or decrease) the influence of the concerned data set on the least squares estimates $\hat{\theta}_w$ given by

$$\hat{\theta}_w = [\mathbf{H}^T\mathbf{W}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y}.$$

Noise characteristics and properties of the estimator

The LS estimation does not incorporate any probabilistic or statistical assumptions. However, the statistical attributes of the error function **e**, the square of which is minimized by the LS estimator, have to be defined in order to establish the statistical properties like bias, error covariance, consistency etc. of the LS estimator.

(a) If the noise vector **e** in (3.2.4) is zero mean with known positive definite covariance matrix **R**, that is

$$E[\mathbf{e}] = 0, \qquad E[\mathbf{e}\mathbf{e}^T] = \mathbf{R}, \qquad (3.2.9)$$

and if **e** and **H** are statistically independent, the estimated parameters will be unbiased as shown below.

Let the parameter estimation error be given by

$$\tilde{\theta} = \theta_o - \hat{\theta}$$

where

$$\mathbf{y} = \mathbf{H}\theta_o + \mathbf{e}, \qquad \theta_o, \text{ the true parameters,}$$

$$\hat{\theta} = [\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{y}, \qquad \hat{\theta}, \text{ the LS estimate.}$$

The orthogonality between **e** and **H** is implicit with the LS parameter estimation. $\tilde{\theta}$ can be expressed as

$$\begin{aligned}\tilde{\theta} &= \theta_o - [\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T[\mathbf{H}\theta_o + \mathbf{e}] \\ &= -[\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{e}.\end{aligned}$$

Hence

$$E[\tilde{\theta}] = -[\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T\, E[\mathbf{e}] = 0.$$

Hence the estimated parameters are unbiased, that is

$$E[\hat{\theta}] = E[\theta_o].$$

The covariance of the parameter estimation error is given by

$$\begin{aligned} E[\tilde{\theta}\tilde{\theta}^T] = \mathbf{P} &= E[[\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{e}\mathbf{e}^T\mathbf{H}[\mathbf{H}^T\mathbf{H}]^{-1}] \\ &= [\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}\mathbf{H}[\mathbf{H}^T\mathbf{H}]^{-1}. \qquad (3.2.10) \end{aligned}$$

Thus the error covariance matrix **P**, which is indicative of the performance of the estimator, does not depend on the observations **y**.

Remark: Simple unbiasedness of the parameters is not very meaningful; the variance should be low too.

(b) If the elements of the noise vector **e** are also Gaussian white with *identical* variances σ^2, that is

$$\mathbf{R} = \sigma^2\mathbf{I},$$

and hence

$$\mathbf{P} = \sigma^2(\mathbf{H}^T\mathbf{H})^{-1}, \qquad (3.2.11)$$

the least squares estimate is unbiased as well as consistent, and is the same as the *best linear unbiased estimate* (BLUE) and the maximum likelihood estimate.

Note that the estimate $\hat{\theta}$ is said to be consistent, if it attains the value θ_o asymptotically, that is if

$$\lim_{N\to\infty} \text{trace}\ \sigma^2(\mathbf{H}^T\mathbf{H})^{-1} = 0,$$

which follows from (3.2.11) in the present case; the trace of a square matrix **A** with elements a_{ij} is given by trace(**A**) = Σa_{ii}.

A consistent estimate need not be unbiased.

3.3 LS ESTIMATION: COMPUTATIONAL ASPECTS

There are mainly two approaches to off-line LS estimation:
(i) solving normal equations, and
(ii) orthogonal LS estimation.

3.3.1 Solving Normal Equations

θ may be estimated directly by solving the normal equation (3.2.7):

$$\hat{\theta} = [H^T H]^{-1} H^T y.$$

The direct solution suffers from poor numerical stability, particularly against round-off errors, because of the explicit inversion $[H^T H]^{-1}$. This inversion can be avoided by using *Cholesky factorization*. For any symmetric positive definite matrix $H^T H$, the Cholesky factorization is given by

$$\begin{aligned} H^T H &= LDL^T \\ &= [LD^{1/2}][LD^{1/2}]^T \\ &= GG^T, \end{aligned} \tag{3.3.1}$$

where **L** is a lower triangular matrix with unity diagonal elements, **D** is a diagonal matrix with positive elements, and **G** is a lower triangular matrix. The estimation procedure is as follows.

(a) Compute $f = H^T y$.

(b) Compute $H^T H$ and perform Cholesky factorizaton:

$$H^T H = GG^T,$$

yielding the normal equations

$$GG^T \hat{\theta} = f, \tag{3.3.2}$$

G is a lower triangular matrix and invertible, assuming **H** to be invertible.

(c) Define $z = G^T \hat{\theta}$, and solve for **z** in

$$Gz = f. \tag{3.3.3}$$

(d) Solve for $\hat{\theta}$ in

$$G^T \hat{\theta} = z. \tag{3.3.4}$$

This approach of estimation belongs to a class called the *square root algorithms*.

Remarks

(1) **G** being triangular, equations (3.3.3) and (3.3.4), can be solved without performing explicit inversion of **G**. For example, from $Ax = y$, **x** can be solved as follows:

```
(a) if A is lower triangular:
    $x(1) = y(1)/A(1,1)$          ! x and y being n vectors.
    For $i = 2$ to $n$
    $x(i) = \{y(i) - \sum_{j=1}^{i-1} A(i,j)x(j)\}/A(i,i);$
(b) if A is upper triangular:
    $x(n) = y(n)/A(n,n)$
    For $i = 1$ to $n-1$
    $x(i) = \{y(i) - \sum_{j=1}^{n-i+1} A(i,n-j+1)x(n-j+1)\}/A(i,i);$
```

(2) Nonuniqueness of square root factors of a matrix does not influence the estimation through (3.3.3) and (3.3.4). For example,

$$\mathbf{G}\mathbf{G}^T = \mathbf{G}\mathbf{W}\mathbf{W}^T\mathbf{G}^T = [\mathbf{G}\mathbf{W}][\mathbf{G}\mathbf{W}]^T,$$

for any $\mathbf{W}$, where $\mathbf{W}\mathbf{W}^T = \mathbf{I}$.

3.3.2 Orthogonal LS Estimation

The use of orthogonal decomposition for LS estimation has been widely studied (Lawson and Hanson, 1974, Golub and Van Loan, 1989). These approaches are extremely well conditioned numerically, and can lead to robust estimation, although at the cost of relatively increased computation. Orthogonal LS estimation using QR decomposition (Appendix 3B) and Singular Value Decomposition (Sec.7.6) are discussed in this section.

Estimation using QR decomposition

The mxn regression matrix $\mathbf{H}$ is decomposed as

$$\mathbf{H} = \mathbf{Q}\mathbf{R},$$

where the columns of the mxm matrix $\mathbf{Q}$ are orthonormal (i.e. $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$), and mxn matrix $\mathbf{R}$ is upper triangular and invertible. Hence

$$\mathbf{H}^T\mathbf{H} = \mathbf{R}^T\mathbf{Q}^T\mathbf{Q}\mathbf{R} = \mathbf{R}^T\mathbf{R}. \tag{3.3.5}$$

The normal equation (3.2.7) becomes

$$\mathbf{R}^T\mathbf{R}\hat{\theta} = \mathbf{R}^T\mathbf{Q}^T\mathbf{y}.$$

So the LS estimate $\hat{\theta}$ is obtained by solving

$$\mathbf{R}\hat{\theta} = \mathbf{Q}^T\mathbf{y}. \tag{3.3.6}$$

Unlike SVD, QR decomposition cannot be used for rank deficient LS estimation problems, that is when **H** is not a full rank matrix; however, QR with column pivoting factorization may be used in such cases as discussed in the next section.

Remarks

(1) Equations (3.3.1) and (3.3.5), show magnitude-wise equivalence between $\bar{\mathbf{R}}$ the upper triangular part of **R** of QR decomposition and **G** of Cholesky factorization as $\mathbf{G} = \bar{\mathbf{R}}^T$.

(2) As a comparison of numerical stability, if the desired computational precision is $<\varepsilon^2$ for the direct solution of the normal equation (3.2.7), the required precision with orthogonal approach is $<\varepsilon$ only, although the latter entails a computational load, almost twice that of the former.

Estimation using singular value decomposition (SVD)

SVD is one of the most robust tools for LS estimation. SVD is discussed in detail in Sec.7.6. The present discussions are confined to the LS estimation only.

SVD of any mxn matrix, **H**, is given by

$$\mathbf{H} = \mathbf{USV}^T, \tag{3.3.7}$$

where mxm **U** and nxn **V** are orthogonal matrices: $\mathbf{U}^T\mathbf{U} = \mathbf{UU}^T = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{VV}^T = \mathbf{I}$; **S** is diagonal with nonincreasing elements (referred to as singular values), ordered down the diagonal:

$$\mathbf{S} = \begin{bmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_n \end{bmatrix}, \qquad s_1 \geq s_2 \geq \ldots \geq s_n \geq 0.$$

Using (3.3.7) on the normal equations,

$$[\mathbf{H}^T\mathbf{H}]\hat{\theta} = \mathbf{H}^T\mathbf{y},$$

$$\mathbf{VSU}^T\mathbf{USV}^T\hat{\theta} = \mathbf{VSU}^T\mathbf{y},$$

or

$$\mathbf{V}^T\hat{\theta} = \mathbf{S}^{-1}\mathbf{U}^T\mathbf{y}.$$

Hence the LS estimate is given by

$$\hat{\theta} = \mathbf{VS}^{-1}\mathbf{U}^T\mathbf{y}. \tag{3.3.8}$$

S being a diagonal matrix, the elements of $\mathbf{S}^{-1}$ are the inverse of the corresponding elements of **S**. Thus the computation in (3.3.8) effectively involves matrix multiplications only.

The unique feature of LS estimation using SVD is that **H** need not be a full rank matrix. LS estimation involves two steps (3.3.7) and (3.3.8). Estimation in case of rank deficient **H** is discussed in the following section.

Remark: **U** need not be explicitly formed for computing $\hat{\theta}$ in (3.3.8), as $U^T y$ can be directly used in the implementation; this can reduce computational load.

Example 3.3.2 Estimate the parameters of the model

$$y(k) = a_o + a_1x_1(k) + a_2x_2(k) + a_3x_3(k) + a_4x_4(k) + e(t),$$

given the following data

Table 3.3.2 Cement curing data (Hald, 1952, Sec.20.3)

Observations	y	x_1	x_2	x_3	x_4
1	78.50	7.0	26.0	6.0	60.0
2	74.30	1.0	29.0	15.0	52.0
3	104.30	11.0	56.0	8.0	20.0
4	87.60	11.0	31.0	8.0	47.0
5	95.90	7.0	52.0	6.0	33.0
6	109.20	11.0	55.0	9.0	22.0
7	102.70	3.0	71.0	17.0	6.0
8	72.50	1.0	31.0	22.0	44.0
9	93.10	2.0	54.0	18.0	22.0
10	115.90	21.0	47.0	4.0	26.0
11	83.80	1.0	40.0	23.0	34.0
12	113.30	11.0	66.0	9.0	12.0
13	109.40	10.0	68.0	8.0	12.0

These data concern the relation between the heat evolved during the hardening of certain cements (y) and four dependent variables: x_1, the percentage of tricalcium aluminate, x_2, the percentage of tricalcium silicate, x_3, the percentage of calcium aluminium ferrate, and x_4, the percentage of dicalcium silicate.

Here **H** is a 13x5 matrix, where the elements of the first column are all 1's (corresponding to the parameter a_o), and the values of x_1 to x_4 are contained in the next four columns of **H**. The estimation of the parameters θ in $y = H\theta + e$ using different approaches is presented here.

Exercise 1: Estimation using Cholesky factorization

$$\mathbf{f} = \mathbf{H}^T\mathbf{y} = [0.1241 \quad 1.0032 \quad 6.2028 \quad 1.3982 \quad 3.4733]^T \times 10^4,$$

$$\mathbf{G} = \begin{bmatrix} 3.6056 & & & & \\ 26.9030 & 20.3772 & & & 0 \\ 173.6212 & 12.3215 & 52.4774 & & \\ 42.4346 & -18.2859 & 1.1199 & 12.5172 & \\ 108.1665 & -14.2316 & -54.6073 & -12.8688 & 3.4497 \end{bmatrix},$$

$$\mathbf{z} = [344.0528 \quad 38.0799 \quad 34.7532 \quad 3.1295 \quad -0.4970]^T.$$

The parameters estimated from (3.3.4) are given by

$$\hat{\theta} = [62.4054 \quad 1.5511 \quad 0.5102 \quad 0.1019 \quad -0.1441]^T.$$

The sum of residual-square = 47.8636.

Exercise 2: Estimation using QR decomposition

The component **R** in **H** = **QR** is given by

$$\mathbf{R} = \begin{bmatrix} -3.6056 & -26.9030 & -173.6212 & -42.4346 & -108.1665 \\ & 20.3772 & 12.3215 & -18.2859 & -14.2316 \\ & & -52.4774 & -1.1199 & 54.6073 \\ & & & -12.5172 & 12.8688 \\ 0 & & & & -3.4497 \\ & & & & \end{bmatrix}.$$

Here **Q** is a 13x13 matrix. The parameters estimated using (3.3.6) are the same as obtained through the Cholesky factorization above. Note the magnitude-wise equivalence between the upper triangular part of **R** and $\mathbf{G}^T$.

Exercise 3: Estimation using SVD

SVD of **H** produces: $\mathbf{H} = \mathbf{USV}^T$, where the singular values are

211.3675, 77.2361, 28.4597, 10.2674, 0.0349,

and

$$\mathbf{V} = \begin{bmatrix} 0.0170 & -0.0037 & 0.0000 & 0.0110 & -0.9998 \\ 0.1279 & 0.0428 & -0.6459 & 0.7513 & 0.0103 \\ 0.8397 & 0.5092 & -0.0181 & -0.1876 & 0.0103 \\ 0.1984 & -0.0721 & 0.7557 & 0.6199 & 0.0105 \\ 0.4888 & -0.8565 & -0.1067 & -0.1263 & 0.0101 \end{bmatrix}.$$

Here U is a 13x13 matrix. The parameters obtained using (3.3.8) are the same as in the earlier cases.

Exercise 4: Estimation with normalized data

If the data columns are normalized with respect to unity mean and variance (for example for column elements y_i:

$$y_i(\text{normalized}) = 1.0 + (y_i - y_{mean})/(y_{max} - y_{min}),$$

H_n becomes as follows.

$$H_n = \begin{bmatrix} 1.0000 & 0.9769 & 0.5077 & 0.6964 & 1.5556 \\ 1.0000 & 0.6769 & 0.5744 & 1.1700 & 1.4074 \\ 1.0000 & 1.1769 & 1.1744 & 0.8016 & 0.8148 \\ 1.0000 & 1.1769 & 0.6188 & 0.8016 & 1.3148 \\ 1.0000 & 0.9769 & 1.0855 & 0.6964 & 1.0556 \\ 1.0000 & 1.1769 & 1.1521 & 0.8543 & 0.8519 \\ 1.0000 & 0.7769 & 1.5077 & 1.2753 & 0.5556 \\ 1.0000 & 0.6769 & 0.6188 & 1.5385 & 1.2593 \\ 1.0000 & 0.7269 & 1.1299 & 1.3279 & 0.8519 \\ 1.0000 & 1.6769 & 0.9744 & 0.5911 & 0.9259 \\ 1.0000 & 0.6769 & 0.8188 & 1.5911 & 1.0741 \\ 1.0000 & 1.1769 & 1.3966 & 0.8543 & 0.6667 \\ 1.0000 & 1.1269 & 1.4410 & 0.8016 & 0.6667 \end{bmatrix}.$$

The singular values of H_n are

s_1 to s_5: 8.0652, 1.6726, 1.3962, 0.4420, 0.0217,

and the estimated parameters are

$$\hat{\theta} = [47.2865 \quad 31.0221 \quad 22.9575 \quad 1.9363 \quad -7.7793]^T.$$

The sum of residual-square (= 47.8636) is the same as with unnormalized data.

Remarks: In the last exercise, since the columns of H_n are normalized, it appears from $\hat{\theta}$ that the variables x_3 and x_4 are relatively insignificant in this problem. If x_3 and x_4 are rejected, the singular values of the truncated (i.e. 13x3) H_n are {6.3264, 0.9911, 0.6832} and the estimated parameters are $\hat{\theta} = [36.2557 \ 29.3661 \ 29.8013]^T$.

3.3.3 Rank Deficient LS Estimation

Rank deficient LS estimation refers to H being rank deficient in the estimation problem

$$\mathbf{y} = \mathbf{H}\theta + \mathbf{e}.$$

As discussed in Sec.3.2.2, the rank deficiency of **H** can be due to collinearity between different regressors, or due to the different sets of data constituting the rows of **H** being too steady or unchanging. When **H** is rank deficient, there are an infinite number of solutions to the LS estimation problem. There are two possibilities for the size of θ:

(i) it can be considered to be n, where **H** is mxn, (m>n),

(ii) it can be considered to be r, the rank of **H**;

SVD can be used in the former case, while subset selection followed by SVD can be used in the latter.

In the present case, there are two basic problems:

(1) determination of the rank of **H**, and

(2) the solution of the LS estimation problem for the full or the truncated parameter-vector.

Rank of a matrix

Singular value decomposition (see Sec.7.6) provides the most direct and definite method for the determination of the rank of a matrix. SVD of an mxn matrix (m>n) **H** produces n singular values arranged in non-increasing order:

$$s_1 \geq s_2 \geq \ldots \geq s_r \geq \ldots \geq s_n \geq 0.$$

Here,

(i) the number (r≤n) of nonzero, that is non-negligible, singular values will be indicative of the rank (r) of the matrix;

(ii) the smallest nonzero or non-negligible singular value gives the distance of the matrix **H** from the set of all (further) rank deficient matrices.

It is implied that if the rank of **H** is r (<n), the singular values $s_{r+1},\ldots,s_n$ are all zero or negligibly small compared with s_r. So rank determination will require declaration of the tolerance δ, where

$$s_r > \delta \geq s_{r+1} \geq \ldots \geq s_n \geq 0.$$

Remarks

(a) If mxn matrix **H** is of rank r (r<min(m,n)), i.e. if **H** is noninvertible, its nxm *pseudo-inverse* is given by

$$\mathbf{H}^{\dagger} = \mathbf{V}\mathbf{S}^{\dagger}\mathbf{U}^{\mathrm{T}},$$

where mxn matrix $S^{\dagger} = \text{diag}(1/s_1, \ldots, 1/s_r, 0, \ldots, 0)$. If the rank of $H = n$, then $H^{\dagger} = (H^TH)^{-1}H^T$. If the rank of H $= m = n$, $H^{\dagger} = H^{-1}$.

(b) Rank of a matrix can also be determined using URV decomposition cited by Stewart (1992); this method is not explored in this book.

LS estimation using SVD

The estimation procedure can be summarized as follows:

(1) Perform SVD of **H** and determine r ($r<\min(m,n)$) the rank of **H**, with

$$s_1 \geq s_2 \ldots \geq s_r \quad \text{and}$$

$$s_{r+1} \geq s_{r+2} \geq \ldots \geq s_n \geq 0, \; r \leq n,$$

where $s_{r+1}, \ldots, s_n$ are zero or negligibly small compared with s_r.

(2) Truncate **U**, $\mathbf{S}^{-1}$ and **V** to $\bar{\mathbf{U}}$, $\bar{\mathbf{S}}^{-1}$, and $\bar{\mathbf{V}}$ of dimensions mxr, rxr and rxn respectively, and

(3) Compute the LS estimate of the n-parameter vector

$$\hat{\theta} = \bar{\mathbf{V}}\bar{\mathbf{S}}^{-1}\bar{\mathbf{U}}^T\mathbf{y}. \tag{3.3.9}$$

Remarks

(a) SVD can provide the solution to the rank deficient LS estimation problem, which is optimum in minimum residual norm sense. Following (3.3.9), the minimum squared residual norm is given by

$$\|\mathbf{y} - \mathbf{H}\hat{\theta}\|_2^2 = \sum_{i=r+1}^{m} c_i^2, \qquad c_i = \mathbf{u}_i^T\mathbf{y},$$

where $\mathbf{U} = [\mathbf{u}_1 \; \mathbf{u}_2 \ldots \mathbf{u}_i \ldots \mathbf{u}_m]$, $\mathbf{u}_i$ being m-column vectors. Although use of SVD ensures the algebraic validity of rank deficient LS solution, the *meaningfulness* of the estimates is not ensured.

(b) An alternative expression for (3.3.9) is given by

$$\hat{\theta} = [\theta_1 \; \theta_2 \ldots \theta_n]^T = \sum_{i=1}^{r} (c_i/s_i)\mathbf{v}_i,$$

where $\mathbf{v}_i$ are the columns of the truncated $\bar{\mathbf{V}}$, that is the first r n-column vectors of **V**:

$$\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_i \ldots \mathbf{v}_r \ldots \mathbf{v}_n], \qquad \bar{\mathbf{V}} = [\mathbf{v}_1 \ldots \mathbf{v}_i \ldots \mathbf{v}_r].$$

LS estimation through subset selection

In case of collinearity, the redundant columns of **H** should be eliminated before parameter estimation, otherwise erroneous parameters will be estimated even if SVD is used for solving the LS estimation problem. Subset selection (Sec.3.6.2 and 3.6.4) provides an easy method for the elimination of redundant regressors in the case of collinearity. Here, the selection of a subset of an information set (i.e. **H**) is discussed; subset selection in regression problems (i.e. with **y** taken into consideration) is discussed in Sec.3.6.4.

If **H** is rank deficient, SVD followed by QR with column pivoting (QRcp) factorization may be used to estimate an r-vector (r being the rank of **H**) LS estimate of the parameters.

The basic idea is to selectively extract an mxr subset $\bar{\mathbf{H}}$ from the mxn **H** through subset selection, discussed in Sec.3.6.2. Now the LS estimation problem is approximated as follows:

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\boldsymbol{\theta} + \mathbf{e} \\ &\approx \bar{\mathbf{H}}\bar{\boldsymbol{\theta}} + \mathbf{e}', \end{aligned} \tag{3.3.10}$$

where $\bar{\mathbf{H}}$, which consists of r columns of **H**, is a full rank matrix. The LS estimate of the reduced r-vector $\bar{\boldsymbol{\theta}}$ is determined by solving (3.3.10) using SVD.

Example 3.3.3 Compute the parameters of the model

$$y(k) = a_1x_1(k) + a_2x_2(k) + a_3x_3(k) + a_4x_4(k),$$

given the following data

Table 3.3.3 Synthetic data

y	x_1	x_2	x_3	x_4
1.2	3.3	0.4	3.5	2.0
0.4	4.5	-1.5	4.6	1.5
1.0	5.4	-0.9	5.6	2.3
1.3	3.6	0.3	3.6	2.0
2.0	5.1	-4.9	5.3	0.2
0.5	6.0	-1.9	6.1	2.1
1.3	2.7	0.7	2.8	1.7
2.4	3.6	-5.0	3.4	-0.8

Here the 8x4 data matrix **H** is given by the columns of x_1, x_2, x_3, x_4. SVD of **H** produces the singular values:

s_1 to s_4: 18.5603, 6.7683, 0.2431, 0.0851

So the rank of **H** is effectively 2. **U** and **V** are given by

$$\mathbf{U} = \begin{bmatrix} -0.2415 & -0.3184 & -0.3526 & 0.3666 & 0.2548 & -0.1152 & -0.3151 & 0.6357 \\ -0.3552 & -0.0885 & 0.2039 & -0.7315 & 0.4059 & -0.3304 & -0.0838 & 0.0916 \\ -0.4150 & -0.2563 & -0.1128 & -0.1632 & -0.8246 & -0.1840 & -0.0943 & -0.0040 \\ -0.2575 & -0.3252 & 0.3483 & 0.4546 & 0.1880 & -0.2977 & -0.2249 & -0.5704 \\ -0.4562 & 0.4761 & -0.6308 & 0.0579 & 0.1638 & -0.0746 & 0.0963 & -0.3495 \\ -0.4708 & -0.1256 & 0.1708 & -0.0436 & 0.0801 & 0.8480 & -0.0697 & -0.0349 \\ -0.1893 & -0.3290 & 0.0155 & 0.1278 & 0.0843 & -0.0802 & 0.9035 & 0.0977 \\ -0.3338 & 0.6071 & 0.5222 & 0.2750 & -0.1209 & -0.1551 & 0.0620 & 0.3594 \end{bmatrix},$$

$$\mathbf{V} = \begin{bmatrix} 0.0027 & -0.5905 & 0.4622 & -0.6616 \\ -0.6696 & -0.1525 & 0.5266 & 0.5012 \\ 0.2909 & -0.7715 & -0.3063 & 0.4758 \\ -0.6834 & -0.1813 & -0.6444 & -0.2911 \end{bmatrix}.$$

Exercise 1:

(a) Using (3.3.8), the estimated parameters work out to be

$\hat{a}_1 = 0.6888$, $\hat{a}_2 = 1.1330$, $\hat{a}_3 = 0.0409$, $\hat{a}_4 = 0.8320$.

The estimated $\hat{y}$ is obtained as

$\hat{y} = [1.2053\ 0.34\ 1.015\ 1.3027\ -1.9888\ 0.4821\ 1.3529\ -2.3809]^T$

and the MSE = 0.0009.

(b) Since only two singular values are large, the parameters may be estimated using (3.3.9). The result is as follows.

$\hat{a}_1 = 0.0600$, $\hat{a}_2 = 0.5420$, $\hat{a}_3 = 0.0792$ $\hat{a}_4 = 0.3107$.

The estimated $\hat{y}$ is obtained as

$\hat{y} = [1.3136\ 0.2876\ 0.9946\ 1.2853\ -1.8677\ 0.4661\ 1.2915\ -2.4732]^T$

and the MSE = 0.0062.

Exercise 2

Since only 2 singular values are large, subset selection may be used to select the two significant rows of **H**. So subset selection is performed on $\bar{\mathbf{V}}^T$, consisting of the 1st 2 columns of **V**. QRcp factorization produces the permutation matrix

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Hence the 2nd and the 3rd columns of **H** are selected (i.e. variables x_2 and x_3 are selected). So the problem can be solved using (3.3.10), where $\bar{\mathbf{H}}$ is an 8x2 matrix. SVD of $\bar{\mathbf{H}}$ (= $\mathbf{U}'\mathbf{S}'\mathbf{V}'^T$) produces the singular values:

$$s_1 = 13.8217, \text{ and } s_2 = 5.2735.$$

The parameter set, given by $\mathbf{V}'\mathbf{S}'^{-1}\mathbf{U}'^T\mathbf{y}$, is determined as

$$\hat{a}_2 = 0.6957, \text{ and } \hat{a}_3 = 0.2920.$$

The estimated $\hat{y}$ is obtained as

$$\hat{y} = [1.3001 \; 0.2995 \; 1.0089 \; 1.2597 \; -1.8613 \; 0.4592 \; 1.3044 \; -2.4856]^T$$

and the MSE = 0.0063.

Remarks

(1) In this simulation example, the data are generated so that x_3 is almost equal to x_1, and x_4 is the approximate average of x_2 and x_1. So the approximate rank of **H**, detected as 2, is correct.

(2) Here, y is in fact noise corrupted y^*, where the elements of y^* are given by

$$y^* = 0.3x_1 + 0.7x_2.$$

It can be verified that if $y = y^*$, where

$$y^* = [1.27 \; 0.30 \; 0.99 \; 1.29 \; -1.90 \; 0.47 \; 1.30 \; -2.42]^T,$$

the parameters work out to be $\hat{a}_1 = 0.3$, $\hat{a}_2 = 0.7$, $\hat{a}_3 = 0.0$ and $\hat{a}_4 = 0.0$, irrespective of the values of x_3 and x_4, provided the problem is not ill-conditioned. For example if one or two singular values are negligibly small, even with $y = y^*$, incorrect parameters will result from (3.3.8); for correct estimation either (3.3.9) has to be used or subset selection has to be performed and then (3.3.10) has to be solved. The result with subset selection will be more meaningful as the model will not be overparameterized.

(3) The subset selection in Exercise 2 above selects x_3 instead of x_1, because the energy $\Sigma x_3^2(k)$ is greater than $\Sigma x_1^2(k)$. It is desirable that the columns are normalized before subset selection; in such a case the regressor vector

most strongly correlated with y should be selected first in QRcp factorization (see Sec.3.6.4 and Appendix 3B).

3.3.4 Estimation with Orthogonalized Regressors

An LS estimation problem: $\mathbf{y} = \mathbf{H}\theta + \mathbf{e}$, is ideally formulated, if the independent variables, that is the regressors given by the columns of **H**, are mutually orthogonal, but are not orthogonal to the vector of dependent variable **y**. In such cases, the estimated parameters are independent of the model order.

Note that orthogonalization and orthogonal transformation are discussed in detail in Chapter 7.

Orthogonality

Two signals $z_i(k)$ and $z_j(k)$ are mutually orthogonal, if

$$\sum_{i=1}^{N} z_i(k)z_j(k) = \begin{cases} 0 & \text{for } i \neq j, \\ C & \text{for } i = j, \end{cases} \quad C \text{ being a constant.}$$

Orthogonal polynomial regression

Consider the nth order polynomial

$$y(k) = \sum_{i=1}^{n} \theta_i x_i(k) + e(k), \qquad k = 1,\ldots,m, \tag{3.3.11}$$

where x_i are the independently appearing variables, y is the dependent variable and e is the error term.

Using matrix notations

$$\mathbf{y} = \mathbf{X}\theta + \mathbf{e}, \tag{3.3.12}$$

where

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(m) \end{bmatrix}, \quad \mathbf{X} = [\mathbf{x}_1 .. \ \mathbf{x}_i \ .. \ \mathbf{x}_n], \quad \mathbf{x}_i = \begin{bmatrix} x_i(1) \\ \vdots \\ x_i(m) \end{bmatrix},$$

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e(1) \\ \vdots \\ e(m) \end{bmatrix}.$$

No sequencing is assumed in **y** or **H**. Let the orthogonalized expression for the independent variables **X** be given by **Z**:

$$\mathbf{y} = \mathbf{Z}\beta + \mathbf{e}; \tag{3.3.13}$$

that is $\mathbf{Z}\beta = \mathbf{X}\theta$, where

$$\mathbf{Z} = [\mathbf{z}_1 .. \ \mathbf{z}_i \ .. \ \mathbf{z}_n], \qquad \mathbf{z}_i = \begin{bmatrix} z_i(1) \\ \vdots \\ z_i(N) \end{bmatrix}, \qquad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}.$$

Equation (3.3.13) can be restated as

$$y(k) = \sum_{i=1}^{n} \beta_i z_i(k) + e(k), \qquad k = 1,\dots,m. \tag{3.3.14}$$

It is implied that the mutually orthogonal set of vectors $\{\mathbf{z}_1 \dots \ \mathbf{z}_i \dots \ \mathbf{z}_n\}$ are produced from the linearly independent set of vectors, $\{\mathbf{x}_1 .. \ \mathbf{x}_i \ .. \ \mathbf{x}_n\}$ so that for p = 1,...,n, the set $\{\mathbf{z}_1 .. \mathbf{z}_i \ .. \ \mathbf{z}_p\}$ spans the same p-dimensional subspace as the set $\{\mathbf{x}_1 \dots \ \mathbf{x}_i \dots \ \mathbf{x}_p\}$. The polynomials (3.3.14) are called *orthogonal polynomials*. The set, $\{\mathbf{z}_1 \dots \ \mathbf{z}_i \dots \ \mathbf{z}_n\}$, constitutes a nonsingular linear transformation of the set, $\{\mathbf{x}_1 \dots \ \mathbf{x}_i \dots \ \mathbf{x}_n\}$, which is the maximum number of linearly independent polynomials over the range k = 1,...,m.

Implementation .using SVD

The SVD of an mxn matrix **A** is given by $\mathbf{A} = \mathbf{USV}^T$, where $\mathbf{U} = [\mathbf{u}_1,..\mathbf{u}_i,..,\mathbf{u}_m]$, $\mathbf{V} = [\mathbf{v}_1,..\mathbf{v}_i,..,\mathbf{v}_n]$, $\mathbf{u}_i$ and $\mathbf{v}_i$ being m and n vectors respectively; the diagonal matrix **S** contains the singular values: $s_1 \geq s_2 \geq \dots \geq s_p \geq 0$, $p = \min(m,n) = n$.

A can be orthogonalized to **Z** as

$$\mathbf{Z} = \mathbf{AV} = \mathbf{US}, \tag{3.3.15}$$

$$\mathbf{Z} = [\mathbf{z}_1,..,\mathbf{z}_i,..,\mathbf{z}_n],$$

where $\mathbf{z}_i$ are m-column vectors of the mxn matrix **Z**. Thus

$$\begin{aligned} \mathbf{A} &= \mathbf{USV}^T = \mathbf{ZV}^T \\ &= \sum_{i=1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T = \sum_{i=1}^{n} \mathbf{z}_i \mathbf{v}_i^T. \end{aligned} \tag{3.3.16}$$

Hence the LS estimation problem given by

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\theta + \mathbf{e} \\ &= \mathbf{ZV}^T\theta + \mathbf{e} = \mathbf{Z}\beta + \mathbf{e}, \end{aligned} \tag{3.3.17}$$

where $\beta \triangleq \mathbf{V}^T\theta$. The LS estimation of β is given by

$$\hat{\beta} = \mathbf{S}^{-1}\mathbf{U}^T\mathbf{y}. \tag{3.3.18}$$

If **H** is not a full rank matrix, β can be estimated as

$$\beta' = \bar{S}^{-1}\bar{U}^T y.$$

where $\bar{S}$ and $\bar{U}$ are the truncated rxn and mxr matrices respectively, r being the rank of **H**; β' is an n-vector.

Implementation using QR factorization

The QR factorization of mxn matrix **H** is given by

$$\mathbf{H} = \mathbf{QR}$$

where mxn **Q** has orthonormal columns and **R** is upper triangular; **R** will be invertible if the columns of **H** are linearly independent, that is if **H** is a full rank matrix.

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\theta + \mathbf{e} = \mathbf{QR}\theta + \mathbf{e} \\ &= \mathbf{Q}\alpha + \mathbf{e}, \qquad \alpha = \mathbf{R}\theta; \end{aligned}$$

the m-parameter vector α is estimated as

$$\alpha = \mathbf{Q}^T\mathbf{y}, \qquad \text{since } \mathbf{Q}^T\mathbf{Q} = \mathbf{I}.$$

This approach requires **H** to be a full rank matrix.

Applications

Two direct applications of orthogonal LS estimation follow.
(1) *Synthesis of signals*: Following (3.3.14) and (3.3.16 -3.3.17),

$$\hat{\mathbf{y}} = \sum_{i=1}^{m} \mathbf{z}_i \hat{\beta}_i, \tag{3.3.19}$$

where $\mathbf{z}_i$ columns are orthogonalized. So given a set of orthogonal vectors (or polynomials) $\mathbf{z}_i$, the vector (or signal sequence) **y** may be synthesized with appropriate *choices* of β.
(2) *Analysis of signals*: Once the LS estimate $\hat{\beta}$ is computed, it remains valid for any model order, that is for any chosen set of regressors $\mathbf{z}_i$ in (3.3.19). In other words, any number of terms $\mathbf{z}_i\beta_i$ may be added to reconstruct $\hat{\mathbf{y}}$, without having to re-estimate $\hat{\beta}$, unlike the case of non-orthogonal LS estimation where for every change of the model order or $\{\mathbf{z}_i\}$ sequence, the parameters θ have to be re-estimated.

Remarks

(a) One drawback of orthogonal LS estimates is that when the information matrix **H** is appended with new information to $m_1$xn matrix $\mathbf{H}_1$, the earlier estimates β cannot be used to compute the corresponding $\hat{y}_1$. $\mathbf{H}_1$ has to be orthogonalized to $\mathbf{Z}_1$ to use β. Alternatively the earlier estimates β have to be retained as $\theta = \mathbf{V}^T\beta$, which can be used directly with $\mathbf{H}_1$.

(b) The physical interpretation of the columns of **H** are lost in the orthogonalization to **Z**.

(c) Different transformations are possible for the orthogonalization of **H**, and the transformed matrix **Z** is not unique.

(d) **Z** together with **V** contains the full information in **H**.

3.4 RECURSIVE LEAST SQUARES METHOD

The least squares method discussed in the last section requires a block of data for estimation, which is a disadvantage, because

(a) at each sampling time or discrete time instant, as a new observation is available, the size of the data block will grow, and hence repeating the LS estimation procedure at each time instant with almost the same data (except for the new observation) will be largely redundant,

(b) often the data are available sequentially only and hence sequential execution of the LS estimation algorithm is preferable.

At any time instant, given the parameter estimates (based on the past data) and the new set of observations, the recursive least squares (RLS) method produces the updated least squares estimates of the parameters.

3.4.1 RLS Formulation

Consider the process model (3.2.1). Given the data on the dependent and the independent variables from the time (k−m+1) to k, the process is described by (3.2.4) and the parameter estimates are given by (3.2.8):

$$\hat{\theta}(k) = [\mathbf{H}^T(k)\mathbf{H}(k)]^{-1}\mathbf{H}^T(k),$$

where the time index k is introduced to signify the least squares estimate being based on the available data up to ·time k.

Now, at time (k+1), with an additional set of data available, the objective is to determine the updated parameters $\hat{\theta}(k+1)$. Following (3.2.4), at time (k+1), the process is given by

$$\begin{bmatrix} y(k) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} H(k) \\ h^T(k+1) \end{bmatrix} \theta(k+1) + \begin{bmatrix} e(k) \\ e(k+1) \end{bmatrix}, \qquad (3.4.1)$$

where

$$h(k+1) = [-y(k)\ -y(k-1) \ldots\ -y(k-N)\ u(k)\ u(k-1) \ldots\ u(k-N)]^T.$$

Similarly to (3.2.8), the least squares estimate is given by

$$\hat{\theta}(k+1) = \left[\begin{pmatrix} H(k) \\ h^T(k+1) \end{pmatrix}^T \begin{pmatrix} H(k) \\ h^T(k+1) \end{pmatrix} \right]^{-1} \begin{bmatrix} H(k) \\ h^T(k+1) \end{bmatrix}^T \begin{bmatrix} y(k) \\ y(k+1) \end{bmatrix}. \qquad (3.4.2)$$

Equation (3.4.2) can be simplified to formulate the recursive least squares estimation law as follows.

$$\hat{\theta}(k+1) = \hat{\theta}(k) + k(k+1)(y(k+1) - h^T(k+1)\hat{\theta}(k)), \qquad (3.4.3)$$

$$k(k+1) = P(k)h(k+1)(1+h^T(k+1)P(k)h(k+1))^{-1}, \qquad (3.4.4)$$

$$P(k+1) = [I - k(k+1)h^T(k+1)]P(k), \qquad (3.4.5)$$

where **k** is the Kalman estimator gain and **P** is the covariance of the parameter-estimation error.

Remarks: k(k+1) given by (3.4.4), can be computed at time k itself, since h(k+1) consists of terms available at time k. Thus at time k+1, as soon as the new measurement y(k+1) is available, $\hat{\theta}(k+1)$ can be produced. This is followed by the updating of P(k+1) and k(k+2).

Derivation of RLS algorithm

Rewriting (3.4.2), dropping most of the arguments and subscripts for the sake of clarity,

$$\begin{aligned} \hat{\theta}(k+1) &= [H^TH + hh^T]^{-1}[H^Ty + hy] \\ &= [H^TH]^{-1}H^Ty + [[H^TH + hh^T]^{-1} - [H^TH]^{-1}]H^Ty \\ &\qquad + [H^TH + hh^T]^{-1}hy. \end{aligned}$$

Since $\hat{\theta}(k) = [H^T H]^{-1} H^T y$,

$$\hat{\theta}(k+1) = \hat{\theta}(k) + [H^T H + hh^T]^{-1}[I - [H^T H + hh^T][H^T H\ ]^{-1}]H^T y$$
$$+ [H^T H + hh^T]^{-1} hy$$
$$= \hat{\theta}(k) + [H^T H + hh^T]^{-1}[-[hh^T][H^T H]^{-1} H^T y + hy]$$
$$= \hat{\theta}(k) + [[H^T H + hh^T]^{-1} h](y - h^T \hat{\theta}(k)),$$

which is the same as (3.4.3) with

$$k(k+1) = [H^T H + hh^T]^{-1} h. \qquad (3.4.6)$$

The recursive expression for **k** is formulated as follows. Consider the *matrix inversion lemma* (Appendix 1):

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}. \qquad (3.4.7)$$

Define the covariance matrix

$$P(k) = [H^T(k)H(k)]^{-1}.$$

Following (3.4.6),

$$k(k+1) = [(P(k))^{-1} + hh^T]^{-1} h \qquad (3.4.8)$$
$$= [P(k+1)]h. \qquad (3.4.9)$$

Using (3.4.7) on the bracketed term in (3.4.8)

$$k(k+1) = [P(k) - P(k)h(1+h^T P(k)h)^{-1} h^T P(k)]h \qquad (3.4.10)$$
$$= P(k)h(1 - (1+h^T P(k)h)^{-1} h^T P(k)h).$$
$$= P(k)h(1 - (1+h^T P(k)h)^{-1}(1+h^T P(k)h-1)).$$

Hence

$$k(k+1) = P(k)h(k+1)(1+h^T(k+1)P(k)h(k+1))^{-1}. \qquad (3.4.11)$$

Again from (3.4.9) and (3.4.10),

$$P(k+1) = P(k) - P(k)h(1+h^T P(k)h)^{-1} h^T P(k).$$

Hence following (3.4.11)

$$P(k+1) = [I - k(k)h^T(k+1)]P(k).$$

3.4.2 Implementation Aspects

The two prime concerns in the implementation of RLS estimation are (a) representativeness of the data, and (b) computational correctness.

Adaptive prediction and control requires the estimator to be able to track the variations in the dynamics of the process. Hence it is necessary to give relatively more importance to the more recent data than the old data, which can be implemented through exponential forgetting of the data in the parameter estimator. It is expected that the more recent data are rich in information. When the data lack information, i.e. when they remain relatively steady or unchanged, they should not be used for parameter estimation.

Again, from a computational point of view the implementation should be numerically well conditioned, and computationally robust, for which algorithms involving square-root updating of the covariance matrix are used.

Exponential forgetting

With exponential forgetting, the RLS estimation laws (3.4.3) to (3.4.5) become

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \mathbf{k}(k+1)(y(k+1) - \mathbf{h}^T(k+1)\hat{\theta}(k)),$$

$$\mathbf{k}(k+1) = \mathbf{P}(k+1)\mathbf{h}(k+1) = \mathbf{P}(k)\mathbf{h}(k+1)(\lambda+\mathbf{h}^T(k+1)\mathbf{P}(k)\mathbf{h}(k+1))^{-1},$$

$$\mathbf{P}(k+1) = \frac{1}{\lambda}\,[\mathbf{I} - \mathbf{k}(k+1)\mathbf{h}^T(k+1)]\mathbf{P}(k), \qquad (3.4.12)$$

minimizing the cost criterion

$$J(\theta,k) = \sum_{i=1}^{k} \lambda^{k-i}\epsilon^2(i), \qquad \epsilon(i) = y(i) - \mathbf{h}^T(i)\theta,$$

where $0 < \lambda \leq 1$. When $\lambda = 1$, the forgetting remains inactive, and all the data are equally weighted. When $\lambda < 1$, the recent data are weighted more than the older data. One meaningful way of specifying the *forgetting factor* λ is through the asymptotic sample length (A_{sl}):

$$\lambda = 1 - \frac{1}{A_{sl}}, \qquad A_{sl} > 1.$$

A_{sl} reflects the number of past samples on which to base the parameter estimation. For a very slowly time-varying process, typically $\lambda = 0.99$; for time-varying processes with stochastic disturbances a higher λ, say $0.95 \leq \lambda < 0.99$, is recommended.

In RLS estimation, it is important that the estimation error covariance matrix **P** is well conditioned. When the process remains relatively steady with little changes in the data, the data vector **h**(k+1) may tend to zero and hence (3.4.12) reduces to

$$\mathbf{P}(k+1) = \mathbf{P}(k)/\lambda.$$

For $\lambda < 1$, an exponential growth of **P**(k) will result, and will cause undesirably large changes in the parameters $\hat{\theta}$, when the data vector again becomes nonzero which may even be due to noise. This phenomenon is known as *covariance windup*. The remedy is to ensure that **P** stays bounded, for which there are various approaches. One way is to stop updating the parameters when $(y(k+1)-\mathbf{h}^T(k+1)\hat{\theta}(k))$, the prediction error, falls too low. Alternatively, the forgetting can be inhibited when the information in the data is low.

Numerical stability and robustness

The covariance updation (3.4.5) is numerically ill conditioned; it is sensitive to computer round-off errors and the differencing operation (between positive terms) in (3.4.5) leads to degradation of computational accuracy.

The numerical stability can be significantly improved by propagating the covariance matrix in the square root form. The basic principle is to factorize the covariance matrix **P** into $\mathbf{RR}^T$, where **R** is the square root of **P**, and to update **R** at every recursion. The alternative is to use $\mathbf{UDU}^T$ factorization due to Bierman (1977); the covariance matrix **P** is factorized as $\mathbf{UDU}^T$, where **D** is a diagonal matrix, and **U** is an upper triangular matrix with 1s on the diagonal. $\mathbf{UD}^{1/2}$ is the square root of **P**. **D** and **U** are propagated through the recursions instead of **P**. Besides the numerical stability, the advantages of $\mathbf{UDU}^T$ factorization are that no explicit square-root extractions are necessary. Sequential propagation of the covariance matrix through U-D factorization is referred to as *U-D covariance measurement updation* which is discussed in Appendix 3.

3.5 SOME SELECTED METHODS: AN INTRODUCTION

Although the least squares method has attractive convergence and asymptotic properties, its main weakness is that the estimates will be biased, if the noise is correlated with measurements of the dependent variable. There are many alternative methods of estimation, a detailed study of which is beyond the scope of this book; this section introduces three such methods belonging to three different classes of parameter estimators.

3.5.1 Instrumental Variable Method

In this method, the parameters are estimated using a set of variables called *instruments* or *instrumental variables*, which are correlated with the regression variables but are uncorrelated with the noise. Consider the process model

$$y(k) + a_1 y(k-1) + \ldots + a_N y(k-N) = b_1 u(k-1) + \ldots + b_N u(k-N) + e(k), \tag{3.5.1}$$

where y is the measured process output and u is the measured process input and e is the error or noise. Restating (3.2.3), the process model (3.5.1) is given by

$$y(k) = \mathbf{h}^T(k)\theta + e(k), \tag{3.5.2}$$

where **h**, the n-data vector, and θ, the n-parameter vector, are given by

$$\mathbf{h}(k) = [y(k-1) \ \ldots \ y(k-N) \ \ u(k-1) \ \ldots \ u(k-N)]^T, \text{ and}$$

$$\theta = [-a_1 \ \ldots \ -a_N \ \ b_1 \ \ldots \ b_N]^T$$

respectively; the noise is not assumed to be uncorrelated with y.

Introduce vector **w**(k) consisting of the instrumental variables

$$\mathbf{w}(k) = [-x(k-1) \ -x(k-2) \ \ldots \ -x(k-N) \ u(k-1) \ \ldots \ u(k-N)]^T, \tag{3.5.3}$$

where x(k-1), x(k-2) etc. are the instruments or instrumental variables, which are chosen so that **w**(k) is uncorrelated with {e(k)} but is strongly correlated with **h**(k). Following (3.5.3) and (3.5.2),

$$\mathbf{w}(k)y(k) = \mathbf{w}(k)\mathbf{h}^T(k)\theta + \mathbf{w}(k)e(k), \tag{3.5.4}$$

using measurements over m samples,

$$\mathbf{y} = \mathbf{H}\theta + \mathbf{e}, \tag{3.5.5}$$

or

$$\mathbf{W}^T\mathbf{y} = \mathbf{W}^T\mathbf{H}\theta + \mathbf{W}^T\mathbf{e} \tag{3.5.6}$$

where **W** is an mxn matrix with rows of $\mathbf{w}^T(k-m+1)$, ..., $\mathbf{w}^T(k)$ and **H** is an mxn matrix with rows of $\mathbf{h}^T(k-m+1)$,..., $\mathbf{h}^T(k)$, and **y** and **e** are m-vectors.

So the instrumental variable estimation follows as

$$\hat{\theta} = (\mathbf{W}^T\mathbf{H})^{-1}\mathbf{W}^T\mathbf{y}, \qquad (3.5.7)$$

since by hypothesis $E[\mathbf{W}^T\mathbf{e}] = \mathbf{0}$, and $E[\mathbf{W}^T\mathbf{H}]$ is positive definite and hence invertible.

The similarity of (3.5.7) with the LS solution leads to the recursive formulation of the instrumental variable method as

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{k}(k)(y(k) - \mathbf{h}^T(k)\hat{\theta}(k-1)), \qquad (3.5.8)$$

where

$$\mathbf{k}(k) = \mathbf{P}(k-1)\mathbf{w}(k)(1 + \mathbf{h}^T(k)\mathbf{P}(k-1)\mathbf{w}(k))^{-1} \qquad (3.5.9)$$

and

$$\mathbf{P}(k) = [\mathbf{W}^T\mathbf{H}]^{-1} = [\sum_{i=1}^{m}\mathbf{w}(i)\mathbf{h}^T(i)]^{-1}$$

$$= \mathbf{P}(k-1) - \mathbf{P}(k-1)\mathbf{w}(k)(1+\mathbf{h}^T(k)\mathbf{P}(k-1)\mathbf{w}(k))^{-1}\mathbf{h}^T(k)\mathbf{P}(k-1). \qquad (3.5.10)$$

Unlike the recursive least squares case, **P** is not a symmetric matrix here.

Choice of instruments

Different choices are possible for the instruments. One of the direct ways of generating the instruments is to derive x(k) from

$$\hat{A}(q^{-1})x(k) = \hat{B}(q^{-1})u(k),$$

where $\hat{A}(q^{-1})$ and $\hat{B}(q^{-1})$ are the estimated polynomials:

$$\hat{A}(q^{-1}) = 1 + \hat{a}_1 q^{-1} + \dots + \hat{a}_N(q^{-N}),$$

$$\hat{B}(q^{-1}) = \hat{b}_1 q^{-1} + \dots + \hat{b}_N(q^{-N}).$$

The parameters $\hat{\theta}$, of $\hat{A}(q^{-1})$ and $\hat{B}(q^{-1})$ polynomials may be generated using any sensible method of estimation. One obvious choice can be to use the RLS estimator, in which case the instrumental variable vector will be given by

$$\hat{\mathbf{w}}(k) = \mathbf{H}^T(k)\hat{\theta}(k).$$

A relatively simpler choice can be to assume $\hat{B}(q^{-1})/\hat{A}(q^{-1})$ introduces a pure lag between u(k) and x(k), leading to

$\hat{w}(k) = [-u(k-\tau)\ -u(k-\tau-1)\ldots\ -u(k-\tau-N+1)\ldots\ u(k-1)\ldots\ u(k-N)]^T.$

It is expected that the correlation between $\hat{w}(k)$ and $h(k)$ will be rendered maximum for a certain value of τ.

The quality of estimation will depend on the way the instruments are generated.

For further discussions on the instrumental variable method and its applications refer to Young (1984), and Söderström and Stoica (1989).

3.5.2 Maximum Likelihood Method

The maximum likelihood method is a powerful and versatile off-line method, which uses the knowledge of the statistical distribution of the observations.

In any estimation problem, there are two basic entities: the data (which may be noisy) and the parameters. The objective of parameter estimation is to obtain the set of parameter values that best conforms to the data. In other words the most likely set of parameters with reference to the data is to be determined; the validity of the parameters is obviously linked with the most likely nature of the data, that is the most probable distribution of the data. If the parameter set is given, and a set of observations is available, the probability density function may be defined for the observations. The problem in the case of parameter estimation is the reverse. Given a probability density function for the distribution of the data, the parameter set may be computed through the maximization of a likelihood function of the parameters with respect to the observations; this is the principle of *maximum likelihood estimation*. The method requires the joint probability density function for the observation being predefined, but it is not restricted to any particular form of the density function.

To estimate the parameters, θ_1, $\theta_2,\ldots$, θ_n, on the basis of the given m observations x_1, $x_2,\ldots$, x_m, a likelihood function $L(\mathbf{x},\theta)$ is introduced.

$$L(x_1,x_2,\ldots,x_m|\theta_1,\theta_2,\ldots,\theta_n) = \prod_{i=1}^{m} f(x_i|\theta_1,\theta_2,\ldots,\theta_n),$$

where $f(x_i|\theta_1,\theta_2,\ldots,\theta_n)$, $i = 1$ to m, is the joint probability density function for the observations: x_i. Let $\mathbf{x}$ and θ stand for the observation vector and the parameter vector respectively. The value of θ_i, for i = 1 to n, which maximizes the likelihood function L is taken as the estimate $\hat{\theta}_i$. Thus the estimate $\hat{\theta}$ is the solution of

$$\frac{\partial L}{\partial \theta_i} = 0, \qquad i = 1,\ldots,n.$$

It is assumed that the joint probability distribution of the possible observations is specified. If there are n parameter values, $\theta_1,\theta_2,\ldots,\theta_n$, that describe the relationship among the observations, and if m number of observations $y = y_1$, $y_2,\ldots, y_m$ are drawn from the specified distribution, let the joint probability density function be denoted by

$$f(x,\theta) = f(y_1,y_2,\ldots,y_m|\theta_1,\theta_2,\ldots,\theta_n). \tag{3.5.11}$$

This is a deterministic function of θ, once the observations are given. It is assumed that the observations are exact although they may not be free from noise contaminations. The function (3.5.11) is called the *likelihood function*, since it is the measure of the likelihood of the observations being valid subject to the choice of the parameters θ. In other words, the estimates of the parameters will be those which maximize the likelihood of the observations y, that is which maximize the likelihood function $f(y|\theta)$.

So the maximum likelihood estimates are obtained as the solutions of

$$\frac{\partial f}{\partial \theta_1} = 0, \frac{\partial f}{\partial \theta_2} = 0, \ldots, \frac{\partial f}{\partial \theta_n} = 0.$$

If $y_1,\ldots,y_N$ are independent observations,

$$\begin{aligned} f(y|\theta) &= f(y_1|\theta)f(y_2|\theta)\ldots f(y_N|\theta). \\ &= \prod_{i=1}^{m} f(y_i|\theta). \end{aligned}$$

Therefore, for mathematical simplification, the logarithmic transformation can be introduced; since a logarithm is a monotonic function of its argument, the value of θ that maximizes $f(x|\theta)$ also maximizes log $f(x|\theta)$. Hence the maximum likelihood estimates $\hat{\theta}$ can be obtained by solving

$$\frac{\partial \log f(y|\theta)}{\partial \theta_i} = 0, \tag{3.5.12}$$

for i = 1 to m.

Remark: The probabilistic characterization of the data may not always be possible in practice, because of the availability of limited data and little control on the generation of the data.

3.5.3 The Koopmans-Levin Method: Implemented using SVD

The estimation methods discussed earlier assume the measurements of the independent variables being noise free. The Koopmans-Levin (KL) method permits noise to be present with both the dependent as well as the independent variables; such classes of LS problems are referred to as *total least squares* problem. This section studies the KL method of parameter estimation, implemented using the singular value decomposition (Fernando and Nicholson, 1985); this method can perform like an approximate maximum likelihood estimator.

Basic principle

Consider an ARMA model for the process:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1), \tag{3.5.13}$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + \ldots + a_Nq^{-N},$$

$$B(q^{-1}) = b_o + b_1q^{-1} + \ldots + b_Nq^{-N}.$$

The variables y and u are available as contaminated measurements y′ and u′ respectively, where

$$y'(k) = y(k) + e_1(k), \qquad \text{and} \quad u'(k) = u(k) + e_2(k);$$

It is assumed that the noise sequences $\{e_1(k)\}$ and $\{e_2(k)\}$ are independent, zero-mean, white noise sequences with known statistical characteristics:

$$E\{e_1(k)\} = 0, \qquad E\{e_2(k)\} = 0.$$

$$E\{e_1(k)e_1(j)\} = \begin{cases} \sigma_1^2 & , k = j \\ 0 & , k \neq j \end{cases}$$

$$E\{e_2(k)e_2(j)\} = \begin{cases} \sigma_2^2 & , k = j \\ 0 & , k \neq j \end{cases}$$

$$E\{e_1(k)e_2(j)\} = 0 \text{ for all } k,j.$$

Define the n-parameter vector

$$\theta = [-1\ -a_1\ \dots\ -a_N\ b_o\ b_1\ \dots\ b_N]^T. \tag{3.5.14}$$

Define the true input-output vector **g**(k) and the observed input-output vector **h**(k) as follows:

$$\mathbf{g}(k) = [y(k)\ y(k-1)\dots\ y(k-N)\ u(k-1)\ u(k-2)\dots\ u(k-N-1)]^T, \tag{3.5.15}$$

$$\mathbf{h}(k) = [y'(k)\ y'(k-1)\ \dots\ y'(k-N)\ u'(k-1)\ \dots\ u'(k-N-1)]^T, \tag{3.5.16}$$

with

$$\mathbf{h}(k) = \mathbf{g}(k) + [\mathbf{e}_1(k)\quad \mathbf{e}_2(k)]^T,$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are the noise vectors. Following (3.5.13 - 3.5.15), $\mathbf{g}^T(k)\theta = 0$. Define the covariance of **h**(k) as

$$R(k) = \sum_{k=1}^{m} \mathbf{h}(k)\mathbf{h}^T(k). \tag{3.5.17}$$

So

$$E\{\mathbf{h}(k)\mathbf{h}^T(k)\} = \lim_{m\to\infty} \frac{1}{m}R(k).$$

Again

$$E\{\mathbf{h}(k)\mathbf{h}^T(k)\} = E\{\mathbf{g}(k)\mathbf{g}^T(k)\} + \begin{bmatrix} E\{\mathbf{e}_1(k)\mathbf{e}_1^T(k)\} & E\{\mathbf{e}_1(k)\mathbf{e}_2^T(k)\} \\ E\{\mathbf{e}_2(k)\mathbf{e}_1^T(k)\} & E\{\mathbf{e}_2(k)\mathbf{e}_2^T(k)\} \end{bmatrix}.$$

Hence

$$E\{\mathbf{h}(k)\mathbf{h}^T(k)\}\theta = E\{\mathbf{g}(k)\mathbf{g}^T(k)\}\theta + \begin{pmatrix} \sigma_1^2 I & 0 \\ 0 & \sigma_2^2 I \end{pmatrix}\theta \tag{3.5.18}$$

$$= \begin{pmatrix} \sigma_1^2 I & 0 \\ 0 & \sigma_2^2 I \end{pmatrix}\theta,$$

since $\mathbf{g}^T(k)\theta = 0$. If $\sigma_1^2 = \sigma_2^2 = \sigma^2$, that is the input and the output noise variances in (3.5.18) are equal (when they are not equal, with appropriate scaling they can be made equal),

$$E\{\mathbf{h}(k)\mathbf{h}^T(k)\}\theta = \sigma^2\theta;$$

so, the parameter vector θ is an eigenvector of the nxn matrix $E\{\mathbf{h}(k)\mathbf{h}^T(k)\}$. The parameter estimation follows through the following arguments:

(i) If the noise variance is significantly small compared with the smallest eigenvalue of $E\{\mathbf{g}(k)\mathbf{g}^T(k)\}$ in (3.5.18), $E\{\mathbf{h}(k)\mathbf{h}^T(k)\}$ will have one significantly small eigenvalue.

(ii) If the noise variance is small, and if $E\{\mathbf{h}(k)\mathbf{h}^T(k)\}$ has one significantly small eigenvalue, θ will correspond to the eigenvector associated with the smallest eigenvalue of $E\{\mathbf{h}(k)\mathbf{h}^T(k)\}$.

Estimation using SVD

The nxn covariance matrix

$$\mathbf{R}(k) = \mathbf{H}^T(k)\mathbf{H}(k),$$

where the mxn matrix

$$\mathbf{H}(k) = [\mathbf{h}(k-m+1)\ \mathbf{h}(k-m+2)\ \ldots\ \mathbf{h}(k)]^T,$$

n being the length of the parameter vector θ. Consider the singular value decomposition of $\mathbf{H}(k)$:

$$\mathbf{H}(k) = \mathbf{USV}^T, \tag{3.5.19}$$

where U and V are orthogonal matrices: $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\mathbf{S} = \text{diag}\ [s_1, s_2, \ldots, s_p]$, is the diagonal matrix with $p = \min(m,n) = n$, as $m \gg n$. Hence

$$\mathbf{R}(k) = \mathbf{VS}^2\mathbf{V}^T. \tag{3.5.20}$$

So, the smallest eigenvalue of $\mathbf{R}(k)$ is given by the smallest diagonal element in $\mathbf{S}^2$: s_n^2, and the corresponding eigenvector is the last column of $\mathbf{V}$. In other words, the parameter vector θ is given by the last column of $\mathbf{V}$, obtained from the SVD of $\mathbf{H}(k)$, and hence $\mathbf{R}(k)$ need not be explicitly formed.

Example 3.5.3 Given the noisy input and output data, estimate the parameters of the simulated process

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + e(k),$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} = 1 - 1.5q^{-1} + 0.7q^{-2},$$

$$B(q^{-1}) = b_o + b_1q^{-1} = 1 + 0.5q^{-1},$$

and $u(k)$ is generated as $(1-0.9q^{-1})^{-1}e(k)$, where $e(k)$ is Gaussian white noise with unit variance.

Table 3.5.3 Estimated parameter values

Exercise	a_1	a_2	b_0	b_1
1	-1.536	0.7147	1.064	0.1891
2	-1.585	0.7492	0.847	0.3182
3	-1.521	0.7444	1.176	0.5076
4	-1.468	0.6767	0.989	0.6151
5	-1.525	0.7117	1.078	0.3237
6	-1.533	0.7389	0.726	0.7200
7	-1.461	0.6561	1.206	0.3101
8	-1.523	0.7050	1.074	0.3584
9	-1.524	0.7265	0.896	0.5912
10	-1.596	0.7716	1.606	0.2736
Average	-1.527	0.7195	1.0662	0.4207
Std.dev.	0.040	0.0328	0.2285	0.1659
True value	-1.500	0.7000	1.0000	0.5000

Here,

$$y(k) + a_1y(k-1) + a_2y(k-2) - b_ou(k-1) - b_1u(k-2) = e(k).$$

The values of the variables on LHS constitute the columns of **H**. Assuming 200 sets of data being available, **H** is 200x5 matrix. SVD of **H** is produced and the last column of 6x6 matrix **V** is noted; the column vector normalized by the first column element gives the parameter values.

The exercise is repeated 10 times with different sets of the data. Irrespective of large additive input and the output noise, the results obtained are reasonably close to the true parameter values as shown in Table 3.5.3.

Remark: This example is taken from Fernando and Nicholson (1985).

Recursive estimation

Define the information matrix

$$\mathbf{P}(k) = \mathbf{R}^{-1}(k).$$

Hence using (3.5.17)

$$\mathbf{P}^{-1}(k) = \mathbf{P}^{-1}(k-1) + \mathbf{h}(k)\mathbf{h}^T(k). \tag{3.5.21}$$

Again following (3.5.20)

$$\mathbf{P}(k) = \mathbf{V}\mathbf{S}^{-2}\mathbf{V}^T.$$

So the lowest eigenvalue of **R**(k) is the same as the largest eigenvalue of $\mathbf{P}(k)$ which will be $1/s_n^2$, and the corresponding column of $\mathbf{V}$ will give the parameter vector. Let

$$\mathbf{P}(k) = \mathbf{C}^T(k)\mathbf{C}(k),$$

and the SVD of $\mathbf{C}(k)$ is given by $\mathbf{C}(k) = \mathbf{U}_c\mathbf{S}_c\mathbf{V}_c^T$,

$$\begin{aligned}\mathbf{C}^T(k)\mathbf{C}(k) &= \mathbf{V}_c\mathbf{S}_c^2\mathbf{V}_c^T \\ &= \mathbf{P}(k) = \mathbf{V}\mathbf{S}^{-2}\mathbf{V}^T.\end{aligned}$$

Since the eigenvalues of a matrix are unique, the diagonal elements of $\mathbf{S}_c^2$ will be the same as those of $\mathbf{S}^{-2}$; so the parameter vector θ will be given by the first column of $\mathbf{V}_c$ which corresponds to the first and the largest diagonal element in $\mathbf{S}_c^2$.

Noting the similarity between (3.5.21) and the equations (3.4.8–3.4.9), the sequential update of $\mathbf{C}(k)$ can be formulated through the U-D covariance measurement update algorithm discussed in Appendix 3A.

Summary

(1) Recursively update $\mathbf{C}(k)$ with the availability of a new data set $\mathbf{h}(k)$ (see Example 3A3, in Appendix 3A).

(2) Compute the SVD: $\mathbf{C}(k) = \mathbf{U}_c(k)\mathbf{S}_c(k)\mathbf{V}_c^T(k)$.

(3) The parameter vector will be given by the first column of $\mathbf{V}_c(k)$ corresponding to the largest singular value in $\mathbf{S}_c$.

Remarks

(a) The identification method studied here, requires the smallest singular value of $\mathbf{H}(k)$ to be distinctly small. More than one singular value of $\mathbf{H}(k)$ being nearly equally small in (3.5.19) causes ambiguity, and hence is undesirable. This can happen, if the noise is not small or if $\mathbf{H}(k)$ is over-parameterized, in which case reparameterization will be required.

(b) The KL method can work with noise corrupted outputs as well as inputs. The noise may not be Gaussian in nature.

3.6 MODEL SELECTION AND VALIDATION

Before the parameters are estimated, the model structure and size have to be specified. Care should be taken in the choice of the model order and in the selection of the specific variables to be incorporated in the model. An overparameterized model is expected to overfit the data; this is because if there is noise associated with the data, an overparameterized model tends to model extraneous noise as well as the information in the data. Such a model lacks in representativeness and shows poor validation against sets of data not used for modelling.

This section starts with a discussion on Akaike Information Criterion, a popular method for model order assessment. Subset selection is studied next. 'Subset selection' is a generic term, meaning selection of specific variables in the candidate set. The procedure for subset selection is different for different problems. Here, first subset selection based on the conventional QR with column pivoting (QRcp) factorization is discussed, which is followed by a case study on best subset AR modelling. Next a modified QRcp factorization scheme is presented for subset selection in a regression problem.

3.6.1 Akaike Information Criterion (AIC)

In identification, any *information criterion* used for selection of the optimal model consists of two components:

(i) a measure of the best model fit and

(ii) a penalty measure on the number of model variables.

Akaike Information Criterion (Akaike, 1974) can be stated as follows. When a model with q independently adjusted parameters is fitted to the data, the AIC of the set S_q is defined as

$$\mathrm{AIC}(S_q) = \log_e(\hat{\sigma}_e^2) + q\gamma, \tag{3.6.1}$$

where $\gamma = 2$ and $\hat{\sigma}_e^2$ is the variance of the residual or the model-fitting error; the model with minimum value of AIC is selected. It is expected that the parameter estimation is based on maximization of the information entropy, as in case of maximum likelihood estimator. For linear regression AIC reduces to

$$\mathrm{AIC}(S_q) = \mathrm{N}\ \log_e\left[\frac{1}{\mathrm{N}}\sum_{k=1}^{\mathrm{N}} e^2(k)\right] + 2q,$$

where N is the number of data inputs, and e is the estimation error: $e(k) = y(k) - \hat{y}(k)$, $\hat{y}(k)$ being the estimate of y(k) based on a model with q parameters. So candidate models of different possible regressor variables are considered and estimated; the model with lowest AIC is selected. AIC may be used with stepwise regression; in stepwise regression (which are also called nested class of models), the model complexity is increased in steps, and AIC may be used for finding the best model. One disadvantage is that there may be many candidate combinations; subset selection can be useful in such cases as discussed in Sec.3.6.3.

Usually AIC tends to overestimate the model order. It was proposed (Bhansali and Downham, 1977) that γ in (3.6.1) may be increased up to 5 to penalize over-parameterization more stringently. There have been many other propositions, e.g., Schwarz (1978) proposed $\gamma = q\log_e(N-q)$. Model order selection criteria are also discussed in Parzen (1974) and Shibata (1985). All the model order selection criteria have a certain degree of inherent subjectiveness, and no particular criterion can be said to be the best.

3.6.2 Subset Selection from an Information Set

Given any mxn information set **A** with m≥n, the objective is to select an mxg subset $\mathbf{A}_1$ (g<n) of **A**, which contains the salient part of the information contained in **A**. In the regression context, **A** is the same as **H** in (3.2.4), where the objective is to select the g significant variables out of the n variables; m indicates the length of the data sets. SVD followed by QRcp factorization has been used for subset selection.

Selection procedure

Let SVD of **A** be given by $\mathbf{A} = \mathbf{USV}^T$, where $\mathbf{U} = [\mathbf{u}_1,\ldots,\mathbf{u}_m]$, $\mathbf{V} = [\mathbf{v}_1,\ldots,\mathbf{v}_n]$, and $\mathbf{S} = [\mathrm{diag}(s_1,\ldots,s_p):0]$, $p = \min(m,n)$. **U** and **V** are the left and the right singular vector matrices respectively. The left and the right singular vectors form a basis for the column-space and the row-space of **A** respectively. Again

$$\mathbf{A} = \sum_{i=1}^{p} \mathbf{u}_i s_i \mathbf{v}_i^T.$$

If g of the p singular values of **A** are dominant, that is s_{g+1}, s_{g+2}, ..., s_p are insignificantly small, the prime information of **A** will be contained in

$$\bar{\mathbf{A}} = \sum_{i=1}^{g} u_i s_i v_i^T.$$

Again

rank(**A**) = the number of nonzero singular values.

So, a selection of $\mathbf{A}_1$, the prime mxg subset of **A**, should correspond to the set of singular values $(s_1,\ldots,s_g)$, $g \leq p$, implying

rank($\mathbf{A}_1$) = pseudorank(**A**) = g.

QRcp factorization can be used for the selection of the subset $\mathbf{A}_1$ as follows.

Let $\bar{\mathbf{V}}$ consist of the first g columns of **V**, that is

$\bar{\mathbf{V}} = [\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_g]$, and let $\bar{\mathbf{V}}^T = [\bar{\mathbf{V}}_1\ \bar{\mathbf{V}}_2]^T$,

where $\bar{\mathbf{V}}_1$ is a gxg and $\bar{\mathbf{V}}_2$ is an (n-g)xg matrix. QRcp factorization, performed on $\bar{\mathbf{V}}$, will produce the nxn permutation matrix **P**, where

$$\mathbf{Q}^T[\bar{\mathbf{V}}_1^T\ \bar{\mathbf{V}}_2^T]\mathbf{P} = [\mathbf{R}_{11}\ \mathbf{R}_{12}],$$

such that $\mathbf{R}_{11}$ is upper triangular and **Q** is a matrix with orthonormal columns. Define matrix $\mathbf{A}_1$ as

$$[\mathbf{A}_1\ \mathbf{A}_2] = \mathbf{AP},$$

where $\mathbf{A}_1$ is an mxg matrix and $\mathbf{A}_2$ is an mx(n-g) matrix; $\mathbf{A}_1$ will have the g prime columns of **A** arranged sequentially in order of decreasing importance (starting from the first column). Thus a dominant subset of **A** is selected.

Remarks

(a) Subset selection will be unique if the (p-q) singular values of **A** are zero. Precise selection of subsets requires a large gap or jump in the distribution of the singular values (i.e. $s_i >> s_{i+1}$, where $1 \leq i \leq p$), otherwise it may not provide sufficient information.

(c) The QRcp factorization on $\bar{\mathbf{V}}^T$ for subset selection is more robust than performing the same on **A**.

Principle of selection

The present selection of columns through QRcp factorization is based on the Euclidean norm. First the column with maximum Euclidean length is selected. Next the column having maximum orthogonal component to the selected column is selected and so on. So the i-th selected column is the one having maximal orthogonal component to the subspace spanned by the earlier selected i-1 columns. The sequence of the selections is stored in the permutation matrix **P**.

The mechanism of pivoting of columns within QRcp factorization is explained in Appendix 3B.

3.6.3 Case Study: Best Subset-AR Modelling using Information Criterion and Subset Selection

For optimal modelling of any time series, both the *number of variables* and the *specific variables* within the model have to be optimally chosen. Any additional term in the model may permit the model to represent *noise*, uncharacteristic of the process, along with the actual underlying process which is undesirable. Suppose a time series $\{y(.)\}$ can be expressed by a full-set AR model of maximal order n. The objective is to identify the best (in terms of minimum AIC or some such information criterion) subset-AR model of order r ($<$n).

The usual procedure is to find AIC exhaustively for all possible models (with all possible combinations of n candidate regressors); the model for which AIC is minimum is considered to be the best subset-AR model. So, for n regressors, (2^n-1) different models have to be considered, which can be computationally expensive. Use of subset selection along with AIC can greatly reduce the domain of exhaustive search for the best subset-AR model.

The problem

For any stationary time series $\{y(.)\}$, the highest lag n for which the partial autocorrelation function of the stationary series is significant is considered as the maximal order of the linear full-set AR model. Let $\{y(.)\}$ be modelled as

$$y(k) = -a_1y(k-1) - a_2y(k-2) -...- a_ny(k-n) + e(k), \qquad (3.6.2)$$

where e is the noise or uncertainty. So following (3.2.1 - 3.2.4) the estimation problem can be expressed as

$$\mathbf{y} = \mathbf{A}\boldsymbol{\theta} + \mathbf{e}, \qquad (3.6.3)$$

where

$$y = [y(n+1)\ y(n+2)\ \ldots y(m+1)]^T,$$

$$A = \begin{bmatrix} y(n) & y(n-1) & \ldots & y(1) \\ y(n+1) & y(n) & \ldots & y(2) \\ & & \vdots & \\ y(m) & y(m-1) & \ldots & y(m-n+1) \end{bmatrix}, \quad m>n,$$

$$\theta = [-a_1\ -a_2\ \ldots\ -a_n]^T,$$

and e is the respective noise vector. Here {y(k)} can be a part of any larger series. The existence of a representative subset-AR model of order r (<n) will show up in one or more of the n singular values of **A** being relatively small.

There are two basic issues in the present identification problem:

(i) The linearly dependent columns of **A** should be eliminated,

(ii) Only appropriate columns of **A** should constitute the linear model (3.6.3); here the appropriateness is decided based on the minimum value of the information criterion.

Information criteria used

The present study uses two different information criteria: AIC and SIC, the latter being the Schwarz Information criterion (Schwarz, 1978). The following normalized forms are used. For any set S_r, with r number of independent variables,

$$AIC(S_r) = \frac{r}{N-r} + \log_e \hat{\sigma}_e^2, \quad \text{and}$$

$$SIC(S_r) = r\log_e(N-r)/(N-r) + \log_e \hat{\sigma}_e^2,$$

where N is the number of stationary observations, r is the order of the full model, and (N-r) is the effective number of observations for fitting the model; $\hat{\sigma}_e^2$, the estimated noise variance, is given by

$$\hat{\sigma}_e^2 = \frac{1}{N-r} \sum_{i=r+1}^{N} \hat{e}_i^2,$$

where e_i are the residuals.

Modelling using subset selection and AIC (or SIC)

The identification of the best subset-AR model can be performed through the following three steps.

(1)(a) Perform SVD of mxn matrix **A** ($\mathbf{A} = \mathrm{USV}^T$). Choose the possible pseudoranks of **A** from the magnitude of the singular values.

(1)(b) For each pseudorank g ($2 \leq g \leq n-1$), perform QRcp factorization on the gxn matrix $\bar{V}^T$ and select the set of relatively independent regressor variables of size g; so the subset S_g of regressors corresponding to each pseudorank g is defined.

(2) For each subset-AR model (corresponding to each pseudorank g), AIC (or SIC) is computed. The subset S_r, corresponding to pseudorank r for which AIC (or SIC) attains the minimum value, is selected; the r regressors of S_r are the candidates for the desired best subset-AR model.

(3) For the (2^r-1) models, with all possible combinations of regressors, AIC (or SIC) is determined. The model producing the minimum value for AIC is the best subset-AR model.

Remark: The consequent reduction of the exhaustive search space for the best subset-AR model is (2^n-1) to (2^r-1), which is computationally advantageous as r can be much smaller than n.

Example 3.6.3(1) Subset-AR model for the sunspot series

The yearly averaged series of the sunspot numbers (Appendix 8A) over the years 1700-1920 are considered for this study. Since the partial autocorrelation is high for the highest lag 9, the full AR model is considered to be of order 9. So the AR model is given by (3.6.2) with n=9. Using the available data, a 213x9 matrix **A** is formed. SVD of **A** produces the singular values (s_1 to s_9):

2053.70654 960.93292 772.59265 293.06665 211.43785
130.61121 98.31714 85.66112 78.89665.

For each pseudorank 8 to 2, subset selection of **A** is performed. The parameters of the AR model corresponding to each subset is now estimated and the AIC and the SIC are computed. The results are shown in Table 3.6.3(1).

Table 3.6.3(1) Subset selection and AIC/SIC for Sunspot series

pseudo- rank g	Set of regressors S(g) (lag of regressor variables)	AIC	SIC
8	6 9 5 7 8 4 2 3	6.29589	6.43886
7	9 1 8 2 5 4 7	5.37993	5.50702
6	9 1 6 4 8 3	5.44842	5.55962
5	9 1 5 3 7	5.45650	5.55181
4	9 1 6 4	5.57438	5.65381
3	9 1 2	5.34758	5.41091
2	8 2	6.79878	6.84628

Since both AIC and SIC are minimum corresponding to the pseudorank 3, an exhaustive search is made in this set to obtain the best subset model, which is found to be

$$y(k) = 1.2495y(k-1) - 0.551y(k-2) + 0.15y(k-9) + e(k), \qquad (3.6.4)$$

with $\sigma_e^2 = 203.261$. Thus in this particular example, the present method *directly* selects the best subset-AR model.

Example 3.6.3(2) Subset-AR model for German unemployment series

365 monthly observations (Jan. 1948 to May 1977) of this series (Appendix 2C) are used for modelling. The partial autocorrelation function shows the maximal order to be 19; for the present exercise the maximal order is overestimated as 20. The series $\{y(k)\}$ is transformed to a relatively stationarity series $\{z(k)\}$, where $z(k) = (1-q^{-1})(1-q^{12})y(k)$, and $\{z(k)\}$ is modelled. Here **A** is a 332x20 matrix. The singular values of **A** are as follows (s_1 to s_{20}):

(0.32460 0.32016 0.30310 0.28957 0.25989 0.24250
0.21911 0.21372 0.20576 0.20087 0.16128 0.15731
0.15257 0.14785 0.14571 0.14445 0.13577 0.13246
0.12687 0.12464) x 10^7

The subsequent subset selection and computation of AIC and SIC values are shown in Table 3.6.3(2).

The minimum AIC is obtained for the pseudorank 12, wheras the minimum SIC is obtained for pseudorank 9.

With pseudorank 12, the best subset AR model is obtained as

Table 3.6.3(2) Subset selection for pseudoranks g, and AIC and SIC for German unemployment series

g	Column numbers	AIC	SIC
19	11 10 12 9 2 19 3 13 20 4 17 15 16 14 7 8 1 18 5	22.9141	23.1433
18	11 10 12 9 2 13 14 19 4 16 15 1 6 7 20 3 8 18	22.9129	23.1306
17	11 10 12 9 2 16 13 8 4 15 14 7 6 20 19 1 17	22.9060	23.1123
16	10 11 12 9 2 15 6 7 14 18 17 16 13 8 20 4	22.9088	23.1036
15	10 11 12 9 2 6 15 8 4 18 1 16 3 17 13	22.8955	23.0789
14	10 11 12 9 7 2 5 16 4 17 15 6 18 8	22.9006	23.0725
13	10 11 9 12 2 5 6 16 3 15 4 14 18	22.8973	23.0577
12	10 11 9 12 2 4 3 8 19 17 1 14	22.8918	23.0408
11	11 10 12 9 2 3 4 8 7 19 18	22.8991	23.0366
10	10 11 9 12 2 4 5 3 8 18	22.8939	23.0200
9	11 10 2 12 7 5 8 13 20	22.8966	23.0112
8	11 2 10 1 7 5 6 4	23.0579	23.1610
7	2 11 10 1 5 15 14	23.0473	23.1390
6	10 1 11 9 2 17	23.0419	23.1222
5	1 20 9 2 12	22.9991	23.0679

$$z(k) = -0.09696z(k-1) - 0.13641z(k-2) + 0.07677z(k-9) - 0.30327z(k-11) - 0.37410z(k-12) + e(k), \qquad (3.6.5)$$

with $\hat{\sigma}_e^2 = 0.818309\text{x}10^{10}$, AIC = 22.8615 and SIC = 22.9302.

With pseudorank 9, exhaustive search produces the best subset-AR model as

$$z(k) = -0.11640z(k-2) - 0.30393z(k-11) - 0.40624z(k-12) + e(k), \qquad (3.6.6)$$

with $\sigma_e^2 = 0.835744\text{x}10^{10}$, AIC = 22.8705, and SIC = 22.9164.

The one-step forecast variance for the two models (3.6.5) and (3.6.6) work out to be $0.499007\text{x}10^9$ and $0.469641\text{x}10^9$ respectively which are very close, although the domain for exhaustive search for the SIC based model is much smaller in this case.

Remarks

(a) It is found (Sarkar and Kanjilal, 1995) that given the maximal order, this method leads to the optimal subset-AR model in terms of the minimum value of the concerned information criterion.

(b) The SIC based approach is found to lead to smaller search space for the best subset-AR model than the AIC based approach.

(c) In the search for S_r, the *minimum* value of the information criterion (IC) may not be taken literally, since no information criterion is absolutely perfect. In the present context, if the IC for a certain subset with lower pseudorank is close to the minimum IC corresponding to a larger pseudorank, the optimal or the near optimal set may lie within the former set. However, the optimal set always lies within the set S_r, for which IC is minimum.

3.6.4 Linear Regression through Subset Selection

In Sec.3.6.2, the selection of a subset from an information set has been discussed. The problem in using this scheme for selection of the prime independent variables in linear regression is that only the candidate regressors are considered here, while their relationships with the output vector **y** cannot be taken into account. To use this subset selection scheme for selection of best set of regressors, some additional procedure (like the use of the information criterion as in Sec.3.6.3) is required to take **y** into consideration. The subset selection scheme is directly applicable for selection of regressors only where the information or data matrix shows a distinct jump in the distribution of its singular values, as in case of Example 3.3.3.

QR factorization incorporating a modified column pivoting scheme for direct successive selection of the most significant regressor variables in order of maximal (mutual) independence as well as correlation with the output vector is presented in this section. Consider the linear modelling problem:

$$y = b_1x_1 + \dots + b_ix_i + \dots + b_nx_n, \tag{3.6.7}$$

where y is the output and x_1 to x_n are the regressor variables. Expressing in concise matrix notation for m sets of data

$$\mathbf{y} = \mathbf{A}\theta, \tag{3.6.8}$$

where $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_i, \dots \mathbf{a}_n]$ is the mxn data matrix containing m vectors of n regressors $\mathbf{a}_i$, and θ is the n-parameter vector. The objective is to select r (<n) most significant variables, leading to the model

$$\mathbf{y} = \bar{\mathbf{A}}\bar{\boldsymbol{\theta}} + \mathbf{e}, \tag{3.6.9}$$

where the mxr matrix $\bar{\mathbf{A}}$ is a subset of $\mathbf{A}$ and $\bar{\boldsymbol{\theta}}$ is the corresponding least squares (LS) estimated r-parameter vector; $\mathbf{e}$ stands for the modelling error or noise. It is assumed that m>n but limited sets of data are available.

Conventional methods

The desired selection of regressors can be performed (a) successively or (b) cumulatively. For successive selection, the regressor with strongest correlation with the output is first selected, and a model with this variable and the output is formed. The consequent residual is computed and the variable having maximum correlation with the residual is selected as the next variable, and once again a model is formed using the selected regressors and so on. Such a method is discussed in Draper and Smith (1968).

For cumulative selection, all candidate models comprising the sets of r variables ($2 \leq r < n$) and the output are formed and the set minimizing a specific statistic is determined; the popular C_p statistic constitutes two additive terms:

$$C_p = RSS_r/s^2 - (m-2r) \tag{3.6.10}$$

where RSS_r and s^2 are the residual sum of squares with r-parameter and the full n-parameter models respectively. Such a method is discussed in Daniel and Wood (1971).

Both the above approaches involve explicit modelling with all candidate sets of regressors and hence the computational requirement is high. The former leads to *optimal successive* selection and the latter leads to the *optimal* selection.

One direct approach for subset selection based on a *modified* QRcp (m-QRcp) factorization method is discussed next.

Fast subset selection using m-QRcp

In m-QRcp factorization based subset selection, the pivoting of the columns is based on the correlation between the rotated output and the rotated candidate variable vectors; the variable showing maximum correlation is selected. The sequence of successive selections is registered in the permutation matrix **P**, where

$$\mathbf{Q}^T\mathbf{A}\mathbf{P} = \mathbf{R}, \qquad \mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_i \ldots, \mathbf{q}_n];$$

$\mathbf{q}_i$ are orthonormal columns, and **R** is upper triangular. The columns of regressor variables in **A** are assumed to be normalized to unit vectors. The successive rotation (leading to the successive selection) of the columns is explained below.

The column vector of **A** producing $\max(\mathbf{a}_i^T\mathbf{y})$ is the most significant one (in successive selection terms), which is swapped with $\mathbf{a}_1$. The **A** so formed is appended by **y** forming $\mathbf{X} = [\mathbf{A}\ \mathbf{y}]$. m-QRcp factorization of **X** is now performed as follows. Using the Gram-Schmidt orthogonalization concept $\mathbf{q}_1$, the unit vector in the direction of $\mathbf{a}_1$ is determined, as $\mathbf{a}_1/\|\mathbf{a}_1\|$. The portion of $\mathbf{a}_j$ (j = 2 to n) and **y** in a direction orthogonal to $\mathbf{a}_1$ will be given by

$$(\mathbf{a}_j - \mathbf{q}_1^T\mathbf{a}_j\mathbf{q}_1) \quad \text{and} \quad (\mathbf{y} - \mathbf{q}_1^T\mathbf{y}\mathbf{q}_1)$$

respectively. This operation is referred to as rotation of $\mathbf{a}_j$ and **y** w.r.to $\mathbf{a}_1$. The selected second vector is the one maximizing

$$(\mathbf{a}_j-\mathbf{q}_1^T\mathbf{a}_j\mathbf{q}_1)^T(\mathbf{y}-\mathbf{q}_1^T\mathbf{y}\mathbf{q}_1),$$

which is swapped with $\mathbf{a}_2$.

At the i-th stage of selection, the rotated variable-vectors ($\mathbf{a}_j^*$) and the rotated output ($\mathbf{y}^*$) vector are

$$\mathbf{a}_j^* = \mathbf{a}_j - (\mathbf{q}_1^T\mathbf{a}_j\mathbf{q}_1 + \ldots + \mathbf{q}_{i-1}^T\mathbf{a}_j\mathbf{q}_{i-1}), \qquad i = 2 \text{ to } n,\ j = i \text{ to } n,$$

$$\mathbf{y}^*= \mathbf{y} - (\mathbf{q}_1^T\mathbf{y}\mathbf{q}_1 + \ldots + \mathbf{q}_{i-1}^T\mathbf{y}\mathbf{q}_{i-1}),$$

and the i-th selected vector is the one maximizing $\mathbf{a}_j^{*T}\mathbf{y}^*$.

The selection is continued for up to r stages. Since the column swappings are recorded in the permutation matrix **P**, $\bar{\mathbf{A}}$ is given by the first r columns of **AP** in (3.6.9). The r parameters $\bar{\theta}$ in (3.6.9) are estimated using the LS method.

An application of this subset selection scheme follows.

Remarks

(1) $\mathbf{a}_i^*$ is in a plane orthogonal to the subspace spanned by earlier (i-1) selected vector spaces.

(2) r can be decided based on the value of C_p in (3.6.10).

(3) The selection is direct and it does not involve any explicit parameter estimation for selection and hence the computational requirement is minimal.

Example 3.6.4 Modelling of rocket engine testing

In this problem the output variable is the chamber pressure (y), and the independent variables are the temperature of the cycle (x_1), the vibration (x_2), the drop shock (x_3), and the static fire (x_4). 24 sets of data are available (Draper and Smith, 1967, p.218), which are as follows

$$y = \begin{bmatrix} 1.4 \\ 26.3 \\ 26.5 \\ 5.8 \\ 23.4 \\ 7.4 \\ 29.4 \\ 9.7 \\ 32.9 \\ 26.4 \\ 8.4 \\ 28.8 \\ 11.8 \\ 28.4 \\ 11.5 \\ 26.5 \\ 5.8 \\ 1.3 \\ 21.4 \\ 0.4 \\ 22.9 \\ 26.4 \\ 11.4 \\ 3.7 \end{bmatrix}, \quad X = \begin{bmatrix} -75 & 0 & 0 & -65 \\ 175 & 0 & 0 & 150 \\ 0 & -75 & 0 & 150 \\ 0 & 175 & 0 & -65 \\ 0 & -75 & 0 & 150 \\ 0 & 175 & 0 & -65 \\ 0 & 0 & -65 & 150 \\ 0 & 0 & 165 & -65 \\ 0 & 0 & 0 & 150 \\ -75 & -75 & 0 & 150 \\ 175 & 175 & 0 & -65 \\ 0 & -75 & -65 & 150 \\ 0 & 175 & 165 & -65 \\ -75 & -75 & -65 & 150 \\ 175 & 175 & 165 & -65 \\ 0 & -75 & 0 & 150 \\ 0 & 175 & 0 & -65 \\ 0 & 0 & -65 & -65 \\ 0 & 0 & 165 & 150 \\ 0 & -75 & -65 & -65 \\ 0 & 175 & 165 & 150 \\ 0 & -75 & -65 & 150 \\ 0 & 175 & 165 & -65 \\ 0 & 0 & 0 & -65 \end{bmatrix};$$

the columns of X are filled with the values of x_1, x_2, x_3 and x_4 respectively. Accomodating nonlinearity in the variables, the engine testing process can be expressed as

$$y = f(x_1, x_2, x_3, x_4, x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4). \tag{3.6.11}$$

The matrix X is extended by the six quadratic terms shown above; let the regressor variables in (3.6.11) be designed as 1 to 10. First, the significant variables are to be selected.

X is appended by the vector y and modified-QRcp factorization is performed. The results are presented in Table 3.6.4 along with the optimum successive selection (through exhaustive search) results. The cumulative squared error (CSE) or RSS is computed based on the least squares estimation of the parameters (including the mean or average term).

Table 3.6.4 Regressor selection and modelling performance in rocket engine testing problem

r	Selection through m-QRcp: selected variables	CSE	Optimal selection: selected variables	CSE
1	4	297.58	4	297.58
2	4 10	157.66	4 10	157.66
3	3 4 10	97.58	2 4 10	70.01
4	2 3 4 10	56.63	2 3 4 10	56.63
5	2 3 4 7 10	53.80	2 3 4 8 10	52.45
6	2 3 4 7 8 10	48.50	2 3 4 7 8 10	48.50
7	2 3 4 7 8 9 10	46.08	2 3 4 7 8 9 10	46.08
8	1 2 3 4 7 8 9 10	44.85	1 2 3 4 7 8 9 10	44.85
9	1 2 3 4 6 7 8 9 10	43.09	1 2 3 4 5 7 8 9 10	41.85

Remarks

(1) The method discussed offers suboptimal (in minimum CSE sense) successive selection of regressors in a linear-in-the-parameter regression problem, quite close to the optimal selection, at a fractional computational cost.

(2) The selection is inherently free from collinearity problems (among the regressor variables), as successive orthogonal subspaces are considered.

(3) In linear regression, successive selection of regressors does not guarantee selection of the optimal (in minimum C_p sense) set, because the regression relationship with optimal selection is a *group phenomenon*. It is possible that two or more variables which are individually weakly correlated with the output constitute a set which is more correlated with the output than the most strongly correlated variable. However, successive selection, being invariably faster than cumulative selection, is worth considering particularly when the number of candidate regressors is large.

3.6.5 Cross Validation

In the cross validation approach, the available data set is divided into two parts; one is used for parameter estimation and the other is used for validation of the estimated models. These two sets are also called the training set and the checking set respectively. Models of increasing complexity may be estimated using the training set and the validation or suitability of the structure is determined based on the minimum of the sum of squared prediction errors between actual output and simulated outputs computed from the test data set using the estimated parameter values. Although cross validation is a useful method, the following points may be noted:

(a) The method is not based on any probabilistic or structural assumptions.

(b) The complete data set is not available for estimation purposes.

(c) There are no definite rules for selecting the number of data points in each set, particularly for short data sets; the different ways of dividing the data set may lead to different results.

Cross validation requires that the estimation and the prediction data should represent the same process dynamics. The data are also expected to be sufficiently rich or informative. Hence, for historical data, which are usually inferior in quality to the experimental data, larger data sets are required. The data splitting methods are discussed in Stone (1974), Snee (1977) and Draper and Smith (1968).

Cross validation is widely used in this book; for example, see (i) COD modelling in Sec.9.4 using GMDH, (ii) COD modelling using a single layer nonlinear model in Sec.9.5, (iii) modelling of COD process and the Mackey Glass series in Sec.10.4.2 and Sec.10.4.3 respectively.

Remark: There exists asymptotic equivalence between the choice of models by Akaike's information criteria and by the cross validation approach, as shown in Stone (1977).

3.7 CONCLUSIONS

Some methods of parameter estimation and linear modelling have been discussed. The emphasis is on the method of the least squares estimation and its sequential implementation. The main attraction of the least squares method is its simplicity of implementation and the ability to produce workable parameter estimates in spite of the noise not being independent of the observations. The estimator shows nice convergence and asymptotic properties, when the noise is independent of the data. For off-line or batch processing, the implementation using orthogonal transformation offers superior numerical properties; in particular SVD based implementation is extremely robust numerically as well as computationally and is applicable even if the estimation problem is ill-conditioned. For the sequential implementation square root algorithms are superior to the general implementations; the U-D filter is well known for its numerical stability and robustness.

An introductory outline of three other methods of estimation: the instrumental variable method, the maximum likelihood method and the Koopmans-Levin method have also been presented; the first two methods are well documented, and are widely used, while the potential of the Koopmans-Levin method has not been fully explored. All these methods can produce unbiased estimates even if the noise is not independent of the observations.

In estimation, model selection is a difficult task. A parsimonious model is desirable as it is likely to lead to a comparatively representative model. AIC or its variants are popularly used for model order selection, but most such methods require exhaustive trials for ultimate selection of the variables. It has been shown that use of SVD and QRcp factorization based subset selection along with AIC or SIC can result in substantial reduction in the domain of exhaustive search for identification of the best subset-AR model. It has also been shown that with a modified column pivoting scheme QR (or m-QRcp) factorization can lead to fast successive selection of regressors in linear regression. Further research is needed in this area.

Cross validation of the identified and estimated model against sets of data not used for modelling, remains viable approach for testing the quality of the model.

REFERENCES

Remarks: The early texts [21,2,11] and the more recent texts [24,31] provide exhaustive coverage of system identification and parameter estimation. The sequential estimation methods are covered in [25,36]. Various aspects of estimation methods and their applications feature in [7,20]. The use of orthogonal decompositions including SVD for solving LS problems are detailed in [14,23]; updating schemes for SVD appear in [8]. Chapter 7 presents a detailed study on SVD. The use of orthogonalized polynomials or regressors feature in [2,13]. Collinearity problems are discussed in [4,32]. Square-root filtering including U-D factorization, and its use in the implementation of LS and RLS problems feature in [6]. The Koopmans-Levin method presented in Sec.3.5.3 is based on [12]; the statistical properties of this method are studied in [3]; the total least squares problems are explored in [18]. AIC, introduced in [1], has been examined in [5,16,26]; AIC and SIC are compared in [22]. Subset selection introduced in [14] has been used with AIC and SIC for subset AR modelling in [27]; selection of best set of regressors in input-output models features in [9,10]; the subset selection scheme presented in Sec.3.6.4 features in [19]. Regression methods have been surveyed in [17]. Cross validation is explored in [30,34,35]; validation methods are also detailed in [20].

[1] Akaike, H. (1974): 'A new look at statistical model identification', *IEEE Trans. on Auto. Control*, AC-19, No.6, 716-723.

[2] Anderson, T.W. (1971): *The Statistical Analysis of Time Series*, John Wiley, New York.

[3] Aoki, M., and P.C. Yue (1970): 'On a priori error estimates of some identification methods', *IEEE Trans. on Auto. Control*, AC-15, No.5, 541-548.

[4] Belsley, D.A., E. Kuh, and R.E. Welsch (1980), *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, John Wiley, New York.

[5] Bhansali, R.J., and D.Y. Downham (1977): 'Some properties of the order of an autoregressive model selected by a generalization of Akaike's FPE criterion', *Biometrika*, 64, 3, 547-51.

[6] Bierman, G.J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[7] Box, G.E.P., and G.M. Jenkins (1976): *Time Series Analysis, Forecasting and Control'*, Holden-Day, San Francisco.

[8] Bunch, J.R., and C.P. Nielson (1978): 'Updating the singular value decomposition', *Numer. Math.*, 31, 111-129.

[9] Daniel, C., and F.S. Wood (1971): *Fitting Equations to Data*, Wiley-Interscience, New York.

[10] Draper, N.R., and H. Smith. (1968): *Applied Regression Analysis*, Wiley, New York.

[11] Eykhoff, P., (1974): *System Identification, Parameter and State Estimation*, John Wiley, London.

[12] Fernando, K.V., and H. Nicholson (1985): 'Identification of linear systems with input and output noise: the Koopmans - Levin method', *IEE Proc. Pt. D.*, 132(1), 30-36.

[13] Freund, R.J., and P.D. Minton (1979): *Regression Methods*, Marcel Dekker, Inc., New York.

[14] Golub, G.H., and C. F. Van Loan (1989): *Matrix Computations*, The Johns Hopkins Univ. Press, Baltimore.

[15] Hald, A. (1952): *Statistical Theory and Engineering Applications*, Wiley, New York.

[16] Hannan, E.J., and B.G. Quinn (1979): 'The determination of the order of an autoregression', *J. Royal Stat. Soc.*, B, 41, No.2, 190-195.

[17] Hocking, R.R. (1983): 'Development in linear regression methodology: 1959-1982', *Technomatrics*, 25(3), 219-249

[18] Huffel, S.V., and J. Vandewalle (1991): *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia.

[19] Kanjilal, P.P., P. Ballav, and G. Saha (1995): 'Fast successive selection of variables in linear models using modified QR factorization', *Electronics Letters*, 31(14), 1204-1205.

[20] Kashyap, R.L., and A.R. Rao (1976): *Dynamic Stochastic Models from Empirical Data*, Academic Press, New York.

[21] Kendall, M.G., and A. Stuart (1961): *Advanced Theory of Statistics*, Griffin, London.

[22] Koehler, A.B., and E.S. Murphree (1988): 'A comparison of Akaike and Schwarz criterion for selecting model order', *Applied Statistics*, 37, 187-195.

[23] Lawson, C.L., and R.J. Hanson, (1974): *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, New Jersey.

[24] Ljung, L. (1987): *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, N.J.

[25] Ljung, L., and T. Söderström, (1983): *Theory and Practice of Recursive Identification*, MIT Press, Cambridge.

[26] Parzen, E. (1974): 'Some recent advances in time series

modelling', *IEEE Trans. on Auto Control*, AC-19, No.6, 723-730.

[27] Sarkar, A., and P.P. Kanjilal (1995): ' On a method of identification of best subset model from full AR model', *Communications in Statistics - Theory and Methods*, 24(6), 1551-1567.

[28] Schwarz, G. (1978): 'Estimating the dimension of a model', *Ann. Statist.*, 6, No.2, 461-464.

[29] Shibata, R. (1985): 'Various model selection techniques in time series analysis', in *Handbook of Statistics*, eds. E.J. Hannan, P.R.Krishnaiah and M.M.Rao, Vol.5, Elsevier Science, Amsterdam, 179-187.

[30] Snee, R.D. (1977): 'Validation of Regression models: methods and examples', *Technometrics*, 19 (4), 415-428.

[31] Söderström, T., and P. Stoica (1989): *System Identification*, Prentice Hall, New York.

[32] Stewart, G.W. (1987): 'Collinearity and least squares regression', *Statistical science*, 2, 68-100.

[33] Stewart, G.W. (1992): 'An updating algorithm for subspace tracking', *IEEE Trans. Signal Processing*, 40(6), 1535-1541.

[34] Stone, M. (1974): 'Cross-validity choice and assessment of statistical predictors', *J. Royal Stat. Soc.*, Ser. B, 36, 111-147.

[35] Stone, M. (1977): 'An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion', *J. Royal Stat. Soc.*, Ser. B, 39, 49-47.

[36] Young, P. (1984): *Recursive Estimation and Time Series Analysis, An Introduction*, Springer-Verlag, Berlin.

CHAPTER 4

SOME POPULAR METHODS OF PREDICTION

Popularly used methods based on time series models or transfer-function models can produce reasonably good predictions often at moderate computational cost.

4.1 INTRODUCTION

Some of the popular methods of the modelling and prediction used by time-series analysts, econometricians and process and business analysts are presented in this chapter. The models used broadly fall into the category of time series models or transfer-function models. These methods are widely covered in the literature, and a detailed study is beyond the scope of this book. The purpose of this chapter is to familiarize the reader with the underlying concepts of some of these methods.

One of the simplest and yet robust methods of prediction is the exponential smoothing based predictor, which is discussed in Sec.4.2. Here the time series is modelled through low-pass filtering (see Appendix 14A). If the data have explicit structural components like a trend or a periodic component etc., the individual structural components may be separately modelled. The procedure has attractively low computational requirement.

The Box and Jenkins method, studied in Sec.4.3, is one of the most popular and powerful methods of prediction. The data are appropriately transformed (through time-differencing etc.) and converted into a stationary series, which is represented by a transfer-function model. The model essentially has two elements: (i) the explicit model of the transformed series, and (ii) the implicit incorporation of process information through the data transformation. Although the computational requirement is moderately high, this method has been successfully applied to a wide variety of processes.

Two other methods briefly discussed in this chapter are the regression method and the econometric method.

4.2 SMOOTHING METHODS OF PREDICTION

Smoothing methods of prediction are largely based on heuristic understanding of the underlying process. Time series with or without seasonal components may be treated. Usually the smoothing methods do not involve any estimation, as the concerned parameters of the model are assumed to be known. Some of the principal smoothing algorithms are presented in this section.

4.2.1 Basic Smoothing Methods

The naive model

The simplest approach to prediction is to disregard any dynamics and assume the variable not to change in future from the present value, i.e. one-step ahead prediction of y(k) given by

$$\hat{y}(k+1)|k) = y(k). \tag{4.2.1}$$

In addition, it may be assumed that the non-zero trend in the time series remains unchanged

$$\hat{y}(k+1|k) = y(k) + (y(k)-y(k-1)). \tag{4.2.2}$$

A time-averaged value for the trend may be used.

Averaging

In simple moving averaging, the averaged value is computed based on the averaging of the data over a moving window (k,(k-n)):

$$\hat{y}(k+1) = \frac{1}{n+1}\sum_{j=0}^{n} y(k-j),$$

typically n=3. Instead of applying equal weighting to the successive data, it may be more meaningful to give more importance to the more recent data.

Linearly increasing weighting may be applied to the data but exponential weighting, referred to as *exponential smoothing*, is more popular in which case

$$\begin{aligned}\hat{y}(k+1) &= \alpha\hat{y}(k) + (1-\alpha)y(k), \qquad 0<\alpha<1 \qquad (4.2.3)\\ &= \frac{(1-\alpha)}{(1-\alpha q^{-1})}\, y(k),\end{aligned}$$

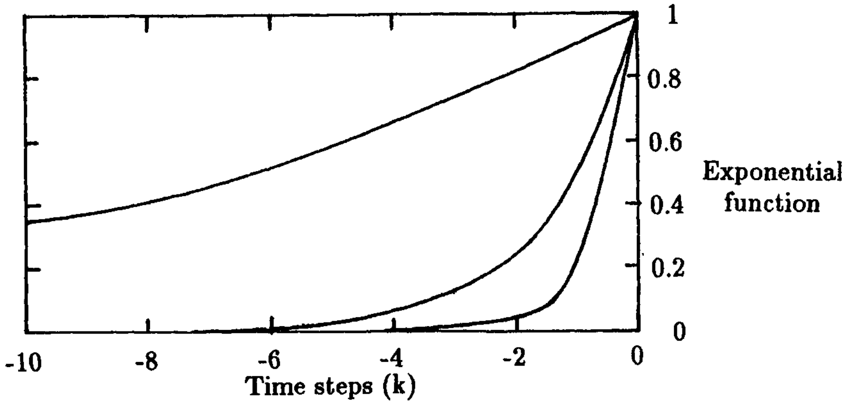


Figure 4.2.1 Exponential weighting function $1/(1-\alpha q^{-1})$, for different values of α; α lies between (0,1).

where on expansion,

$$\frac{1}{1-\alpha q^{-1}} = 1 + \alpha q^{-1} + \alpha^2 q^{-2} + \alpha^3 q^{-3} + \ldots$$

The pattern of weighting, which is exponential, is shown in Fig.4.2.1. The smaller the value of α, the faster the older data are forgotten, or in other words, more importance is ascribed to the more recent data, implying shorter memory for the prediction scheme. For $\alpha = 0$, (4.2.3) reverts to the naive model (4.2.1); $\alpha = 1$ is not permissible. Typical values of α are 0.1 to 0.3.

Remarks

(a) Analogy to low-pass filtering: In terms of frequency domain analysis, the exponential smoother acts as a low-pass filter (see Appendix 14A), where the relatively higher frequency components contained in the data sequence are attenuated. The higher the value of α, the lower are the frequencies passed through.

(b) Lag in the smoothed results: In any data processing, if only past values are used, the processed data tend to lag in time with respect to the original time series. In case of data with seasonal variations, the lag shows up in the conspicuous shifts in the peaks and the troughs of the series. The lag can be reduced by

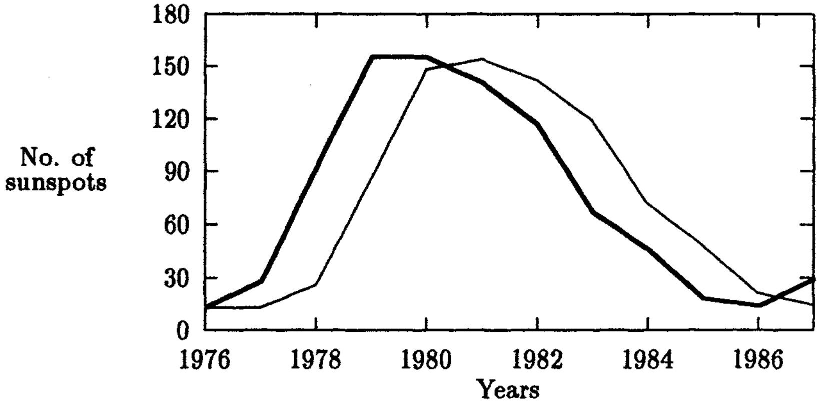


Figure 4.2.2 One-step ahead prediction of the sunspot series using exponential smoothing with $\alpha = 0.1$ (━━━ data, ——— prediction).

(1) using centred averaging or smoothing (as in Appendix 4),
(2) bidirectional filtering (Sec.14.6.1), or
(3) backcasting or backward-smoothing the data prior to use for forecasting or forward-smoothing (Box and Jenkins, 1976, p.199).

One common feature of these three methods is the elimination of the lag, which results from the inherent use of both the past and the future data with respect to the point of smoothing. The first two approaches are discussed elsewhere as stated; in the last approach, the data set is reversed in time and the normal smoothing is performed, the resulting data are time-reversed and the smoothing process is repeated. The result is lag-free smoothing.

Example 4.2.1 Prediction of the sunspot series over short time-span

The modelling of the yearly averaged sunspot series (Appendix 8A) features widely in this book. Here the one-step ahead prediction of the series over twelve years (1977–1987) is considered. The model (4.2.3) is used with $\alpha = 0.1$. The results are presented in Fig.4.2.2. As expected the prediction tends to lag the actual series; the lag increases for higher value of α, where the high frequencies are largely attenuated due to the effect of low-pass filtering of the exponential smoother.

4.2.2 Multiple Smoothing Algorithms

In (4.2.3) only one-parameter smoothing has been performed; instead the series may be considered as a composite of more than one structural component (like the average, the trend and the seasonal component) each of which can be individually modelled.

Predictor models for series without seasonality

Such types of time series can be expressed as

$$y(k) = y_{av}(k) + py_{tr}(k) + e(k), \qquad p=0,$$

where $y(k)$, $y_{av}(k)$, $y_{tr}(k)$ and $e(k)$ are the data, the average (or mean) value, the trend component and the modelling error respectively at time k. Both the average and the trend components are individually modelled using exponential smoothing.

The p-step ahead prediction is given by

$$\hat{y}(k+p|k) = y_{av}(k) + py_{tr}(k).$$

One popular approach is the Holt's method.

Two parameter double exponential smoothing: Holt's method

The average and the trend components are modelled as

$$y_{av}(k) = (1-\alpha)y(k) + \alpha(y_{av}(k-1) + y_{tr}(k-1))$$

$$y_{tr}(k) = (1-\beta)y_{tr}(k-1) + \beta(y_{av}(k) - y_{av}(k-1));$$

α and β lie between (0,1). The predictor may be initialized as

$$y_{av}(1) = y(1), \qquad y_{tr}(1) = \frac{(y(1)-y(0)) + (y(2)-y(1))}{2}.$$

Predictor models for series with seasonality: Winters' method

Here, in addition to the average and the trend components, a seasonal component (y_s) is also considered. The trend and the seasonal components may be linked multiplicatively or additively, which are modelled using exponential relationship (Winters, 1960) as follows.

Multiplicative predictor model

Considering the trend and the seasonal components not to be independent of each other, the model is expressed as

$$y(k+p) = (y_{av}(k) + py_{tr}(k))y_s(k+p-N) + e(k+p);$$

The structural components are initialized and updated as follows.

(a) Initialization of trend and average components:
Let the period length of the series be 12 (e.g., for yearly periodic monthly data). Let the centred average of one complete period preceding and following y(13) be called Y_1 and Y_2 which are given by

$$Y_1 = \frac{1}{2}\left[\left(\frac{y(1)+y(2)+\ldots+y(12)}{12}\right) + \left(\frac{y(2)+y(3)+\ldots+y(13)}{12}\right)\right], \tag{4.2.4a}$$

$$Y_2 = \frac{1}{2}\left[\left(\frac{y(13) + \ldots + y(24)}{12}\right) + \left(\frac{y(14) + \ldots + y(25)}{12}\right)\right]. \tag{4.2.4b}$$

The centre for Y_1 and Y_2 will be y(7) and y(19) respectively. The initial value for the trend component $y_{tr}(13)$ at the 13th month is given by

$$y_{tr}(13) = \frac{Y_2 - Y_1}{12}. \tag{4.2.5}$$

The initial value for the average component $y_{av}(13)$ is given by

$$y_{av}(13) = \frac{Y_2 + Y_1}{2}, \tag{4.2.6}$$

while the same for any other point can be computed using $y_{tr}(13)$ and Y_1 or Y_2; for example, for the 15th month,

$$y_{av}(15) = Y_2 - (19-15)y_{tr}(13).$$

(b) Initialization of seasonal component:
To find the initial value for y_s, first the corresponding average component is extracted; for example, for the i-th month in the period of interest,

$$y_s'(i) = \frac{y(i)}{y_{dc}(i)};$$

the desired initial estimates of the seasonal components are now computed as

$$y_s(i) = y_s'(i) \Big/ \frac{1}{N}\sum_{i=1}^{N} y_s'(i),$$

N being the period length.

(c) Updation of structural components:
The three components y_{av}, y_{tr} and y_s are updated using the principle of exponential smoothing as

$$y_{av}(k) = \alpha(y_{av}(k-1)+y_{tr}(k-1)) + (1-\alpha)(y(k)/y_s(k-N)),$$

$$y_{tr}(k) = \beta(y_{tr}(k-1)) + (1-\beta)(y_{av}(k)-y_{av}(k-1)),$$

$$y_s(k) = \gamma(y_s(k-N)) + (1-\gamma)(y(k)/y_{av}(k));$$

with the chosen smoothing parameters, α, β and γ, lying between (0,1). Usually, γ is higher than α or β, since the seasonal component varies relatively slowly.

(d) Prediction:
The p-step ahead prediction produced at time k is given by

$$\hat{y}(k+p|k) = (y_{av}(k)+py_{tr}(k))y_s(k+p-N).$$

If p>N, instead of $y_s(k+p-N)$, $y_s(k+p-2N)$ may be used which is the best available estimate of the seasonal component concerned.

Additive predictor model

Here, the trend and the seasonal components are assumed to be independent of each other, as follows

$$y(k+p|k) = (y_{av}(k)+py_{tr}(k)) + y_s(k+p-N) + e(k+p-N).$$

The modelling procedure is similar to the multiplicative case. The initial seasonal component is computed as follows:

$$y_s'(i) = y(i) - y_{av}(i), \qquad y_s(i) = y_s'(i) \Big/ \frac{1}{N}\sum_{i=1}^{N} y_s'(i).$$

y_{tr} is initialized the same way as before. The updating is done as follows.

$$y_{av}(k) = \alpha(y_{av}(k-1) + y_{tr}(k-1)) + (1-\alpha)(y(k)-y_s(k-N)),$$

$$y_{tr}(k) = \beta y_{tr}(k-1) + (1-\beta)(y_{av}(k)-y_{av}(k-1)),$$

$$y_s(k) = \gamma y_s(k-N) + (1-\gamma)(y(k)-y_{av}(k)).$$

The multistep prediction is given by

$$\hat{y}(k+p|k) = y_{av}(k) + py_{tr}(k) + y_s(k+p-N).$$

Remarks

(a) Cost and performance: Exponential smoothing is a simple method which is computationally cheap and fast. At the same time this method can perform remarkably well in comparison with other complex methods.

(b) Adaptive prediction: The smoothing coefficients, α, β and γ, may be chosen from prior experience, and kept fixed particularly when predictions for a large number of time series are to be performed, (e.g., sales prediction for commodities in a departmental stores). However, the optimum values of the coefficients may be computed by solving a minimum mean prediction-error square problem, (as in Theil and Wage, 1964), which may be formulated as a usual least squares estimation problem.

4.3 BOX AND JENKINS METHOD

The Box and Jenkins method involves

(a) transformation of the univariate or multivariate time series into stationary time series, and

(b) modelling and prediction of the transformed data using a transfer-function model.

A discrete-time linear model of the time series or process is used. Prior to use, the data series are transformed into stationary series (discussed in Sec. 2.2.1), this is to ensure that the probabilistic properties of mean and variance of the data series remain invariant over time. Usually suitable time-differencing is performed on the data sequences to achieve stationarity. Identification of the model and estimation of the parameters are performed using standard methods.

4.3.1 Modelling Characteristics

The process is modelled as a linear filter driven by a white noise sequence. For example

$$y(k) = e(k) + c_1e(k-1); \qquad (4.3.1)$$

that is $y(k) = (1 + c_1q^{-1})e(k),$

where $\{e(k)\}$ is the discrete white noise sequence and $\{y(k)\}$ is the time series or process output sequence. Here q^{-1}, the backward shift operator, is the same as the backward shift operator B used in the statistics literature, as in

$$By(k) = y(k-1) = q^{-1}y(k), \text{ or say}$$
$$B^d y(k) = y(k-d) = q^{-d}y(k).$$

A generalized model can be expressed as

$$\begin{aligned} &y(k) + a_1y(k-1) + a_2y(k-2) +\ldots+ a_py(k-p) \\ &\quad = e(k) + c_1e(k-1) + c_2e(k-2) +\ldots+ c_re(k-r); \end{aligned} \tag{4.3.2}$$

that is

$$A(q^{-1})y(k) = C(q^{-1})e(k),$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + a_2q^{-2}+\ldots+a_pq^{-p}, \\ C(q^{-1}) &= 1 + c_1q^{-1} + c_2q^{-2}+\ldots+c_rq^{-s}. \end{aligned} \tag{4.3.3}$$

Before the data series are used for modelling, the data may be subjected to

(i) nonlinear transformation, and

(ii) stationarity transformation.

The purpose of nonlinear transformation (discussed in Sec. 8.2.4) is to be able to represent the process by a linear model. The stationarity transformation is to transform the time series into a stationary series through nonseasonal and/or seasonal time-differencing. It may be noted that the data transformation is a way of implicitly incorporating available information into the data used for modelling.

Nonseasonal time-differencing

First time-difference of the series $\{y(k)\}$ is given by

$$\begin{aligned} Y_1(k) &= y(k) - y(k-1) = (1-q^{-1})y(k) \\ &= \Delta y(k), \end{aligned}$$

where Δ is the unit time-difference operator.

A further stage of first differencing applied to the series $\{y(k)\}$, leads to the series,

$$\begin{aligned} Y_2(k) &= Y_1(k) - Y_1(k-1) \\ &= (y(k)-y(k-1)) - (y(k-1)-y(k-2)) \\ &= y(k) - 2y(k-1) + y(k-2). \end{aligned}$$

That is

$$Y_2(k) = (1-q^{-1})^2 y(k)$$
$$= \Delta^2 y(k), \tag{4.3.4}$$

which is the expression for second-order differencing.

Thus dth-order differencing is given by

$$Y_d(k) = (1-q^{-1})^d y(k)$$
$$= \Delta^d y(k), \tag{4.3.4}$$

which results from d successive (unit) time differences being performed on the data sequence. In practice, d, the degree of differencing, rarely needs to exceed 2 in order to be close to a nonstationary process.

A generalized expression for a nonseasonal model is given by

$$A(q^{-1})\Delta^d y(k) = C(q^{-1})e(k), \tag{4.3.5}$$

which is an ARIMA model of order (p,d,r), where the discrete-time polynomials $A(q^{-1})$ and $C(q^{-1})$ are of order p and r respectively.

Remark: In an ARIMA (p,d,r) model (4.3.5), p is the order of autoregressive (AR) part, r is the order of the moving average (MA) part and d is the degree of time-differencing applied to {y(k)}. Hence, to be precise, if any of p, d or r is zero, the model is not ARIMA process any more. So an ARIMA (p,o,o) is in fact an AR(p) process and so on.

Seasonal time-differencing

If the time series {y(k)} is seasonal or periodic in nature with a period length of N, it is expected that the data series will be

(i) seasonally related, (i.e. y(k) related to y(k-N) etc.), besides being

(ii) serially related, i.e. y(k), y(k-1), y(k-2), etc. will be mutually related.

Consider a basic seasonal model

$$y(k) - y(k-N) = e(k); \tag{4.3.6}$$

here the time series {y(k)} is transformed by first degree of seasonal time-differencing:

$$y(k) - y(k-N) = (1-q^{-N})y(k)$$
$$= \Delta_N y(k),$$

where Δ_N $(= 1-q^{-N} = 1-B^N)$ is the seasonal N-difference operator.

If D-degree of seasonal time-differencing is performed on the data, a generalized seasonal model may be expressed as

$$A(q^{-1})(1-q^{-N})^D y(k) = C(q^{-1})e(k). \qquad (4.3.7)$$

In recognition of the serial correlation of the data, if additional d-th order differencing is introduced in (4.3.7), the multiplicative seasonal model or the mixed seasonal and nonseasonal model is obtained.

$$A(q^{-1})(1-q^{-1})^d(1-q^{-N})^D y(k) = C(q^{-1})e(k); \qquad (4.3.8)$$

that is

$$A(q^{-1})z(k) = C(q^{-1})e(k), \text{ with}$$

$$z(k) = \Delta^d \Delta_N^D y(k); \qquad (4.3.9)$$

where the parameters of $A(q^{-1})$ and $C(q^{-1})$ are estimated using the transformed sequence $\{z(k)\}$.

There may be more than one seasonal component in the data. For example, in the case of hourly electrical power load demand series (Appendix 7D), there will be

(a) a daily seasonal component requiring Δ_{24} differencing, and

(b) a weekly seasonal component requiring Δ_{168} differencing. Hence multiple periodicity of order n can be accommodated in the model (4.3.9) by substituting

$$\Delta_{N_1}^{D_1}\ \Delta_{N_2}^{D_2} \ldots\ \Delta_{N_n}^{D_n} \text{ for } \Delta_N^D,$$

where D_1, $D_2, \ldots, D_n$ are usually unity.

Example 4.3.1(1) Assessment of the presence of periodicity in the transformed German unemployment series using SVR spectrum

As discussed in Appendix 11, the presence of a dominant periodicity in the data series is indicated by a repeating peak at multiples of the concerned period length in the SVR spectrum.

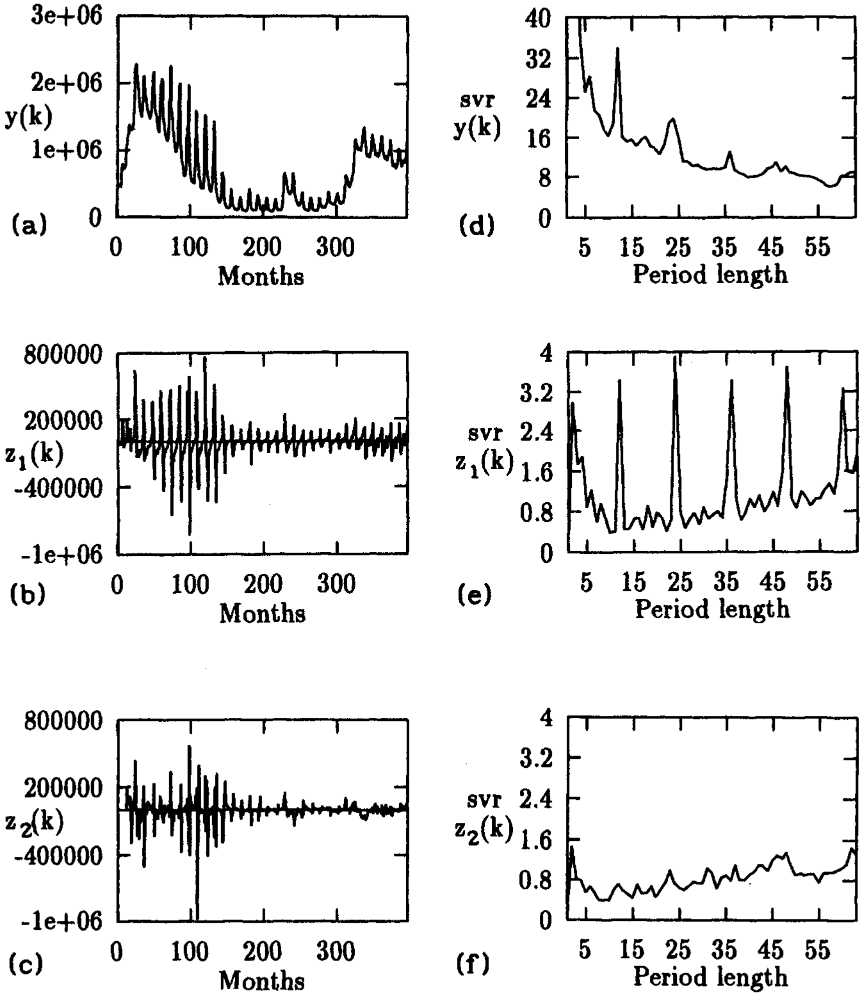


Figure 4.3.1 The original and transformed German unemployment series and their SVR spectra. (a), (b), (c) show $\{y(k)\}$, $\{z_1(k)\}$ and $\{z_2(k)\}$ series, and (d), (e), (f) show their respective SVR spectra.

The monthly German unemploment data series $\{y(k)\}$ (Appendix 7E, Fig.4.3.1a) shows a trend as well as a seasonal component of period length 12. Consider the differencing transformations:

$$z_1(k) = \Delta y(k), \text{ and } \; z_2(k) = \Delta_{12} y(k).$$

Fig.4.3.1 ((a) to (f)) shows the sequences $\{y(k)\}$, $\{z_1(k)\}$ and $\{z_2(k)\}$, and their SVR spectra. Peaks at multiples of the period length 12 are seen to be present in the spectra of both $\{y(k)\}$ and $\{z_1(k)\}$, whereas this peak disappears in case of $\{z_2(k)\}$. This shows that the yearly periodic component present in $\{y(k)\}$ or in $\{z_1(k)\}$ is extracted through Δ_{12} differencing.

Prediction

For a process modelled as

$$A(q^{-1})\Delta^d y(k) = C(q^{-1})e(k), \tag{4.3.10}$$

one-step ahead minimum mean square error prediction is defined as the conditional expectation of $y(k+p)$, at time k, that is

$$\hat{y}(k+1|k) = E(y(k+1)|y(k),y(k-1)\ldots).$$

The error sequence $\{e(k)\}$ may be expressed as

$$e(k) = y(k) - \hat{y}(k|k-1),$$

$$e(k-1) = y(k-1) - \hat{y}(k-1|k-2) \text{ etc.}$$

Once the parameters are estimated, the predictions can be computed using (4.3.10), where all future values of the error term $e(k+1)$, $e(k+2)$ etc. are assumed to be zero, since $E(e(k+1)|k) = 0$.

Example 4.3.1(2) Prediction of an ARIMA (1,1,1) process

Consider the model:

$$(1-a_1q^{-1})\Delta y(k) = (1-c_1q^{-1})e(k).$$

Here,

$$y(k) = y(k-1) + a_1y(k-1) - a_1y(k-2) + e(k) - c_1e(k-1).$$

So

$$y(k+1|k) = (1+a_1)y(k) - a_1y(k-1) + e(k+1|k) - c_1e(k),$$

where $e(k+1|k) = 0$; $e(k)$ may be approximated by

$$e(k) = y(k) - \hat{y}(k|k-1).$$

One-step ahead prediction is given by

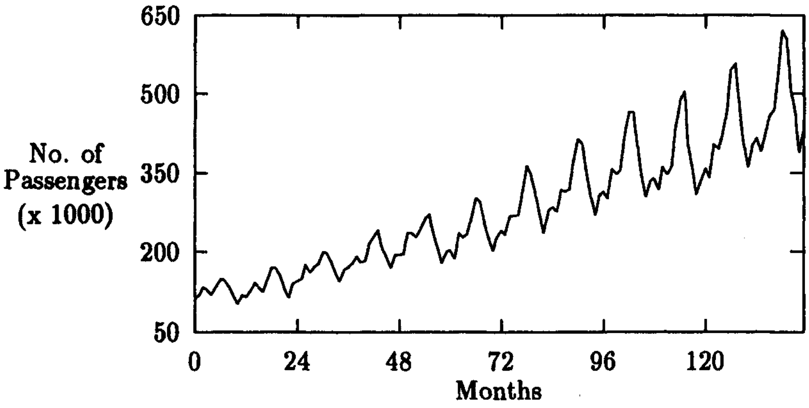


Figure 4.3.2 The airline traffic series (Appendix 7A.2)

$$\hat{y}(k+1|k) = (1+a_1)y(k) - a_1y(k-1) - c_1e(k),$$

where a_1 and c_1 are the estimated parameter values.

Again to compute $\hat{y}(k+2|k)$, form

$$y(k+2|k) = (1+a_1)y(k+1|k) - a_1y(k) + e(k+2|k) - c_1e(k+1|k).$$

So the two-step ahead prediction is given by

$$\hat{y}(k+2|k) = (1+a_1)\hat{y}(k+1|k) - a_1y(k),$$

and so on.

Example 4.3.1(3) AR modelling of the airline traffic series

This series (See Fig.4.3.2 and Appendix 7A.2) contains monthly data over 12 years. The objective is to model the series using the data for the first 9 years and to produce the prediction for the next 3 years.

The series has a trend and a yearly periodic component, and also a diverging pattern. The series is logarithmically transformed to eliminate the divergence; further, unit differencing is applied to eliminate the trend component, and periodic differencing is applied to eliminate the periodic component. Thus the data series {y(k)} is transformed to {z(k)} where

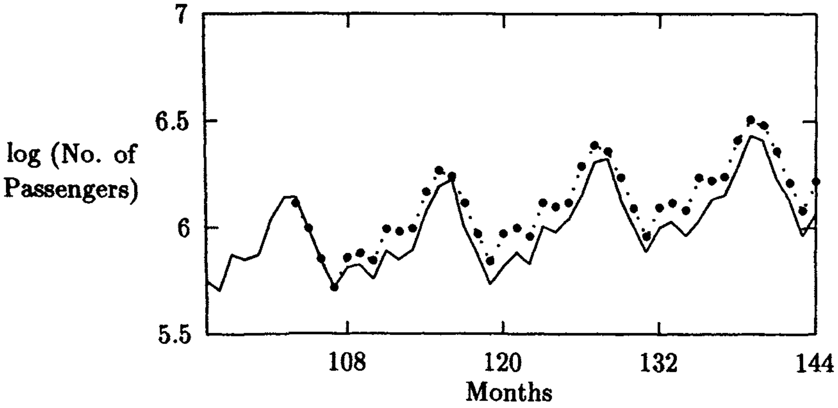


Figure 4.3.3 One- to 41-step ahead prediction of airline series using the subset-AR model (4.3.11b) (•..•..• prediction).

$$z(k) = (1-q^{-1})(1-q^{-12})\log_e y(k).$$

A full AR model for $\{z(k)\}$ can be expressed as

$$z(k) = a_1 z(k-1) + a_2 z(k-2) + \dots + a_n z(k-n) + e(k);$$

here n, the maximal order, given by the lag for which the partial autocorrelation function of $\{z(k)\}$ is reasonably high, works out to be 12. As discussed in Sec.3.6.3, the best subset-AR model can be determined using an information criterion (like AIC) and subset selection, and using the complete data set $\{z(k)\}$, the model works out as

$$z(k) = -0.3004z(k-1) - 0.4205z(k-12) + e(k),$$

with AIC = -6.477, and the residual variance = $1.46\text{x}10^{-3}$; If data up to July of the 9th year are assumed to be available, the model works out as

$$z(k) = -0.326z(k-1) - 0.446z(k-12) + e(k) \qquad (4.3.11a)$$

with AIC = -6.322 and the residual variance = $1.66\text{x}10^{-3}$.

p-step ahead prediction can be computed from

$$\hat{z}(k+p|k) = -0.326z(k+p-1|k) - 0.446z(k+p-12|k),$$

(4.3.11b)

where predicted values (from (4.3.11b)) are used for $z(k+p-12|k)$ and $z(k+p-1|k)$ if the actual data are not available. The one to 41-step ahead prediction result for the series is shown in Fig.4.3.3; the corresponding MSE = 0.0089. The parameters are estimated using the Marquardt algorithm (Box and Jenkins, 1976, p.504).

4.3.2 Implementation Aspects

Some necessary conditions for robust implementation of the formulations presented in Sec.4.3.1 are discussed here.

Stability and invertibility

Consider a generalized model

$$A(q^{-1})z(k) = C(q^{-1})e(k),$$

where $\{z(k)\}$ is the appropriately differenced or filtered observation sequence $\{y(k)\}$. It is necessary that both $A(q^{-1})$ and $C(q^{-1})$ polynomials are stable; $A(q^{-1})$ being stable implies $C(q^{-1})/A(q^{-1})$ is bounded for bounded input $\{e(k)\}$, while $C(q^{-1})$ being stable implies the model is *invertible* (see Appendix 12A for further discussions).

Time-differencing aspects

The differencing of the data sequences to induce stationarity needs to be performed carefully.

(a) When the data are noisy:
A spurious noise contamination will increase on differencing. So unexpected excursions or outliers in the data should be eliminated before differencing. Alternatively the data may be low-pass filtered (e.g., $T(q^{-1})$ filter as in (2.4.7)) or bidirectionally filtered (Sec.14.3), before differencing, to reduce or eliminate the effects of noise.

(b) Over-differencing of the data:
Over-differencing is undesirable, as it leads to *noninvertible* models. For example, if a data series incorporates a deterministic polynomial trend of degree n, nth degree of differencing will eliminate the trend producing a stationary series. Any further differencing will render the model noninvertible (see Sec.8.2.1)

Appropriate degree of differencing can be ensured by the examination of the autocorrelation function, which goes towards zero, as the lag increases. The problem of over-differencing can be avoided, if the roots of $C(q^{-1})$ are constrained to lie inside the unit circle.

Parsimonious parameterization

It is important that the lowest possible model order, or to be more precise, minimum number of variables are used for modelling. Overfitting of a data set by too many parameters makes the model unsuitable for data sets which are not used for modelling, and large modelling errors result. Over-parameterization may incorporate common factors between the left and right hand sides of the expression for the model (e.g., in equations (4.3.5), and (4.3.8)), which may not be precisely identifiable leading to erroneous modelling and poor predictions. In the present context, over-differencing is akin to over-parameterization.

4.4 OTHER SELECTED METHODS

Two other popular methods of prediction are the regression method and the econometric method.

Regression method

The regression method concerns prediction of a dependent (endogenous) variable through the relationship with a number of independent (exogenous) variables, where the disturbances or the uncertainties in the model are statistically defined. Regression methods are problem specific and hence can be of different types, e.g., linear regression, nonlinear regression, multiple linear regression, multivariate regression etc. typically as follows.

Linear regression:

$$y(k) = a_0 + a_1x_1(k) + a_2x_2(k-1) + a_3x_3(k-3) + e(k).$$

Nonlinear regression:

$$y(k) = a_1x(k) + a_1^2x(k-2).$$

Multiple linear regression:

$$\begin{aligned} y(k) = \; & a_1x_1(k) + a_2x_1(k-1) +\ldots+ a_nx_1(k-n) \\ & + b_1x_2(k) + b_2x_2(k-1) +\ldots+ b_nx_2(k-n) \\ & + c_1x_3(k) + c_2x_3(k-1) +\ldots+ c_nx_3(k-n) +\ldots+ e(k). \end{aligned}$$

Mutivariate regression:

$$\begin{aligned} y_1(k) &= a_1x_1(k) + a_{12}x_1(k-1) + \ldots \\ &\quad + b_{11}x_2(k) + b_{12}x_2(k-1) + \ldots + e_1(k), \\ y_2(k) &= a_{21}x_1(k) + a_{22}x_1(k-1) + \ldots \\ &\quad + b_{21}x_2(k) + b_{22}x_2(k-1) + \ldots + e_2(k). \end{aligned}$$

The symbols used have their usual meanings.

Two basic issues arise in regression modelling:
(a) selection of the regressors which are essential for representative modelling of the process concerned, and
(b) the proper estimation of the parameters.

Each selected regressor is expected to be independent of other regressors, and at the same time correlated with the dependent variable. A regressor, which is a linear function of the other variables, is redundant and hence should be eliminated. Regressors which are orthogonal to the output are not necessarily redundant.

For further discussions on regression models see Secs. 3.3.2, 3.3.3, 3.6.4, and 9.3.

Econometric method

Econometrics deal with the study of economic systems or processes in the statistical framework. Usually an econometric model has more than one independent (exogenous) variable and one or more dependent (endogenous) variable. Thus regression methods can be considered to be a subset of econometric methods. However, most economic problems involve relationships between a number of *interdependent* variables, i.e. the dependent variables themselves usualy have *causal* influence on each other (note: the term causal means caused by). Such relationships are expressed by sets of simultaneous equations. The expressions relating interdependent variables through sets of simultaneous equations form a class of models unique to econometrics. The parameters of the simultaneous equation models can be estimated using the two stage least squares or the three stage least squares method (Theil, 1971, Judge *et al* 1982). In practice ordinary least squares are used for approximate solutions.

Econometric models are difference equation expressions which are usually nonlinear and nonstationary in nature. The models are often too large, and usually there are uncertainties associated with the specification of the model equations. This is due to the inadequacy of economic theory to aggregate different economic variables as well as the imprecise knowledge of the response patterns. The limited number of time series observations also cause poor understanding of the underlying process. In practice, for large scale models, the judgemental decision based on the understanding of the process can play a significant role in the predictions.

Example
Based on the Keynesian economic model (named after the economist J.M. Keynes, 1883-1946), the national income can be expressed by the simplified model (Levi, 1983):

$$Y = C + I_0 + (F_{x0} - F_I), \qquad C = C_0 + cY, \qquad \text{and}$$

$$F_I = F_{I0} + mY,$$

where Y is the national output or income (GNP), C is the aggregate consumption of goods, I_0 is the given amount of investment, F_{x0} is the given amount of exports, and F_I is the imports. The C_0 part of consumption does not depend on income. The income dependent propensity to consume and to import are given by c and m respectively. Here the uncertainty or the equation errors have not been shown. For more complete macroeconomic models, see Chow (1981).

Remarks
(1) Econometric models tend to be large in size. It may be necessary to compromise between the prior knowledge of the underlying economic process and the statistical justification coupled with assessment of redundancy.
(2) The noise associated with data may be due to unrepresentative sampling or measurement as well as conceptual misunderstanding.
(3) Economic processes usually have seasonality which may be removed from the data before modelling; both the seasonal and the trend components may be modelled as non-deterministic terms.
(4) The time series forecasting assumes an open-loop architecture, whereas econometric modelling and forecasting assume a closed-loop architecture. The problem of solving simultaneous equations appearing in econometric models is

analogous to that found in the case of multivariable control problems having cross coupling between different inputs and outputs.

4.5 CONCLUDING REMARKS

The main features of some of the methods of modelling and prediction, which are popular with time series analysts and forecasters, have been briefly discussed.

The smoothing methods of modelling are computationally cheap and robust; these are applicable to series with limited dynamics. The low-pass filtering implicit with the smoothing methods result in the elimination of (a) the spurious disturbances which usually appear as outliers, and (b) the time lag in the smoothed data. The time lag problem can be remedied by using centred moving averaging or by using bidirectional smoothing or fixed interval smoothing.

It may be noted that in systems engineering, the term *smoothing* stands for the estimation of the future values of the series $\{y(k)\}$, $\hat{y}(k+i|k)$ where $i \le 0$ (see Sec.14.2), whereas in the present context of time series analysis $i > 0$.

The Box and Jenkins modelling approach involves appropriate transformation of the data to improve the stationarity. Periodic information can be incorporated into the trans- formed series through periodic differencing of the data. There are various ways of transforming the data, and reasonably good predictions can be produced, using the Box and Jenkins method for a variety of processes. Some care is necessary in the data differencing procedure; spurious disturbances, or high frequency noise are exaggerated through the differencing of the data and hence should be removed separately, say by outlier rejection or prior low-pass filtering of the data etc. It may also be noted that in the method of constrained prediction discussed in Chapter 5, the appropriate increments are constrained and thereby the extent of noise remains contained.

The regression model remains one of the popular methods of modelling and prediction. For proper representation of the underlying process, linearly dependent or redundant regressors should be eliminated.

The econometric models are comparatively difficult to construct because of the inherent feedback between the variables of interest. The modelling problem will be simplified, if the influences of different variables can be decoupled.

REFERENCES

Remarks: The smoothing methods proposed in [2,18] and the Box and Jenkins method detailed in [1] are widely explored in many texts [13,14]. The practical aspects are discussed in [7,9,17] and the problems are stated in [3]. Regression methods are treated in [5,8]; a standard text on modelling through linear and nonlinear regression is [6]. Econometric methods of modelling and forecasting are discussed in [16,4,12,10]. Validation aspects feature in many texts, see [11] for a detailed discussion.

[1] Box, G.E.P., and G.M. Jenkins (1976): *Time Series Analysis Forecasting and Control*, Holden-day, San Francisco, CA.

[2] Brown, R.G. (1963): *Forecasting and Prediction*, Prentice-Hall, Englewood Cliffs, N.J.

[3] Chatfield, C., and D.L. Prothero (1973) : 'Box-Jenkins seasonal forecasting: problems in a case-study', *Journal of Royal Stat. Soc.*, Ser. A, 13C, 295-305.

[4] Chow, G.C. (1981): *Econometric Analysis by Control Methods*, Wiley, 1981, New York.

[5] Daniel, C., and F.S. Wood (1980): *Fitting equations to Data*, 2nd edn., Wiley-Interscience, New York.

[6] Draper, N., and H. Smith. (1981): *Applied Regression Analysis*, 2nd edn., Wiley, New York.

[7] Harrison, P.J. (1967): 'Exponential smoothing and short-term sales forecasting', *Management Sciences*, 13, No.11, 821-842.

[8] Huang, D.S. (1970): *Regression and Econometric Methods*, John Wiley, New York.

[9] Jenkins, G.M. (1975): 'Practical experiences with modelling and forecasting time series', in *Forecasting*, Ed. O.D. Anderson, 43-166, North-Holland, Amsterdam.

[10] Judge, G.G., R.C. Hill, W. Griffiths, W.Lütkepwhl, and T.C. Lee (1982): *An Introduction to the Theory and Practice of Econometrics*, Wiley, New York.

[11] Kashyap, R.L., and A.R. Rao (1976): *Dynamic Stochastic Modelling from Empirical Data*, Academic Press, New York.

[12] Klein, L.R., and R.M. Young (1980): *An Introduction to Econometric Forecasting and Forecasting Models*, D.C. Heath, Lexington, Mass.

[13] Makridakis, S., S.C. Wheelwright, and V.E. McGee (1983): *Forecasting Methods and Application*, 2nd edn., John Wiley and Sons, New York.

[14] Montgomery, D.C., L.A. Johnson, J.S. Gardiner (1990): *Forecasting and Time Series Analysis*, 2nd edn., McGraw-Hill, New York.

[15] Theil, H. and S. Wage (1964): 'Some observations on adaptive filtering', *Management Sciences*, 10(2), 198-224.

[16] Theil, H. (1971): *Principles of Econometrics*, Wiley, New York.

[17] Tiao, G.C. (1985): 'Autoregressive moving average models, intervention problems and outlier detection in time series', *in Handbook of Statistics*, Vol.5, 'Time Series in the Time Domain', Eds. E.J. Hannan, P.R. Krishnaiah, M.M.Rao., 85-118.

[18] Winters, P.R. (1960): 'Forecasting sales by exponentially weighted moving averages', *Management Sciences*, 6, 324-342.

CHAPTER 5

ADAPTIVE PREDICTION USING TRANSFER-FUNCTION MODELS

The prediction can be more meaningful when properly constrained, based on process knowledge.

5.1 INTRODUCTION

Adaptive prediction is usually based on minimization of the mean square prediction error. The prediction involves a two stage procedure:

(i) estimation of the parameters of an appropriate model of the time series or the process, and

(ii) reconfiguration of the process model into a prediction model, and computation of prediction using the estimated parameters.

Usually the second stage is trivial and the quality of prediction depends solely on the quality or representativeness of the model.

In practice, the prediction may not always be sensible, some of the reasons for which can be explained as follows.

(a) It is implicitly assumed that the available or measured data are exact, or are corrupted by a white noise sequence. However, if the time series is the measured output of an industrial plant, the desired signal may not be the actual measurement but some underlying process variable. The measurement itself may be contaminated with significant noise of bandwidth overlapping with the signal. The measurement may also be indirect and deduced from other measurements (consider for example, the measurement of the product strength in the iron ore sintering process, discussed in Sec.5.5.1).

The minimum mean square error predictor attempts to track the measured data and hence will be inevitably susceptible to disturbances associated with the data.

(b) The variable being predicted may have distinctive characteristics which should be embedded in the predictor as prior information. For example, the signal may be inherently rate limited (e.g., the movement of a ship or say the monetary inflation of a country). The conventional predictors can incorporate such features in the model only, and not in the prediction algorithm.

For an ideal predictor, it is important that misleading information in the data is ignored and the underlying process variable is followed with due regard to the probable future values, which are based on the user's prior knowledge of the process. Assimilation of the subjective knowledge in the prediction algorithm is expected to result in sensible and meaningful predictions. This chapter presents a constrained mean square prediction error strategy based on penalization of both the prediction error and the increments of prediction; the latter is added to accommodate prior process knowledge into the prediction algorithm.

The commonly used minimum mean square error prediction scheme is discussed in Sec.5.2. The concept of constrained mean square error prediction is explained in Sec.5.3, and the prediction algorithms are produced. A recursive method of extending optimal prediction to multistep predictions is discussed in Sec.5.4. Sec.5.5 is devoted to a case study on the application of adaptive prediction to the iron ore sintering process. This chapter is supported by two appendices; the recursive solution of the Diophantine identity, used for multistep prediction, is presented in Appendix 5A, and the formulation of a multivariable predictor is discussed in Appendix 5B.

5.2 MINIMUM MEAN SQUARE ERROR PREDICTION

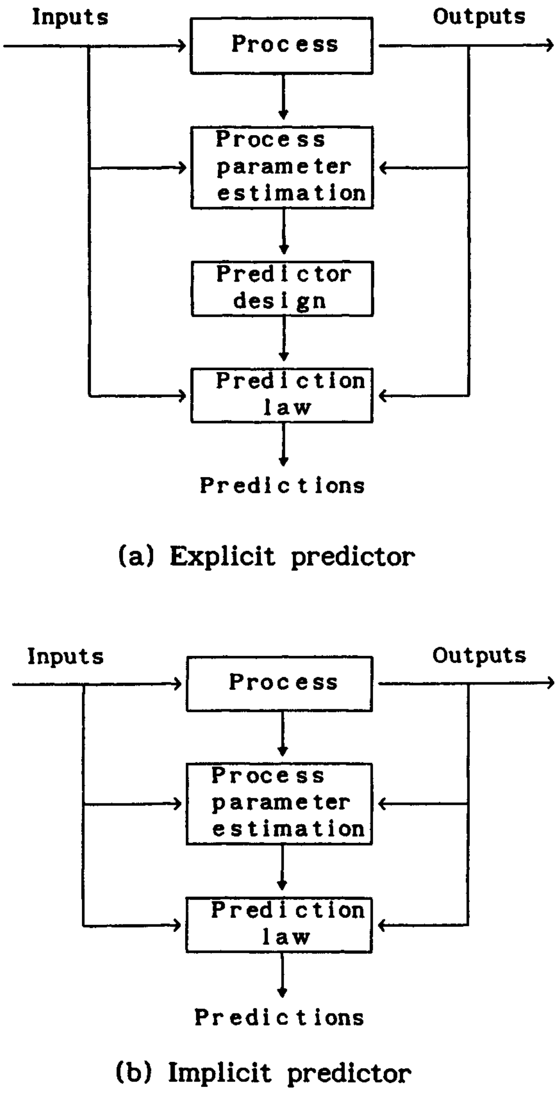
The minimum mean square error prediction is based on the minimization of the cost function

$$J_m = E\{\varepsilon^2(k+p)\},$$

where the prediction error,

$$\varepsilon(k+p) = y(k+p) - \hat{y}(k+p|k),$$

$\hat{y}(k+p|k)$ is the p-step ahead prediction of the output y at time k; the objective is to produce $\hat{y}(k+p|k)$.



(a) Explicit predictor

(b) Implicit predictor

Figure 5.2.1 Adaptive prediction schemes using
(a) Explicit (indirect) approach, and
(b) Implicit (direct) approach.

In the present context, there are two basic methods of prediction (Fig.5.2.1):

(i) *Explicit or indirect method*: The process model is estimated from the time series or the input-output data. A prediction model is formed using the estimated parameters and the multistep predictions are produced.

(ii) *Implicit or direct method*: A predictor model is considered, the parameters of which are directly estimated using the process data; the predictor model implicitly represents the process.

5.2.1 Explicit (Indirect) Prediction

ARMAX model based design

Consider the process model

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k), \tag{5.2.1}$$

where y is the output of the process, u is the known input to the system, and {e(k)} is an uncorrelated random noise sequence. A and B are polynomials in the discrete backward shift operator q^{-1}:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_nq^{-n}, \\ B(q^{-1}) &= b_o + b_1q^{-1} + b_2q^{-2} + \dots + b_nq^{-n}, \\ C(q^{-1}) &= 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_nq^{-n}, \end{aligned}$$

where $q^{-1}y(k) = y(k-1)$. It is assumed that the input and the output sequences are mean square bounded and that the roots of $C(q^{-1})$ lie inside the unit circle.

Remark: The roots of $C(q^{-1})$ lying *inside* the unit circle are equivalent to the roots of the reciprocal polynomial C(q) lying *outside* the unit circle, which means that the solution of $C(q^{-1}) = 0$ should lie inside the unit circle (i.e. $-1<|q|<1$). For example, if $C(q^{-1}) = 1-0.7q^{-1}$, the root is at 0.7, or equivalently $C(q) = 1-0.7q$, having the root at 1/0.7; here $C(q^{-1})$ or C(q) is said to be stable. □□

To compute p-step ahead prediction, define the identity (also referred to as the Diophantine identity):

$$C(q^{-1}) = E_p(q^{-1})A(q^{-1}) + q^{-p}F_p(q^{-1}), \tag{5.2.2}$$

where the degree of $E_p(q^{-1})$, $\delta E_p = p-1$, and $\delta F_p<\delta A$:

$$\begin{aligned} E_p(q^{-1}) &= 1 + e_1q^{-1} + \dots + e_{p-1}q^{-p+1}, \\ F_p(q^{-1}) &= f_o + f_1q^{-1} + \dots + f_{n-1}\, q^{-n+1}. \end{aligned}$$

Following (5.2.1) and (5.2.2),

$$E_p(q^{-1})A(q^{-1})y(k+p) = E_p(q^{-1})B(q^{-1})u(k+p-d) \\ + E_p(q^{-1})C(q^{-1})e(k+p).$$

Using (5.2.2),

$$C(q^{-1})y(k+p) = F_p(q^{-1})y(k) + E_p(q^{-1})B(q^{-1})u(k+p-d) \\ + E_p(q^{-1})C(q^{-1})e(k+p).$$

Hence

$$y(k+p) = \frac{F_p(q^{-1})}{C(q^{-1})}\, y(k) + \frac{E_p(q^{-1})B(q^{-1})}{C(q^{-1})}\, u(k+p-d) \\ + E_p(q^{-1})e(k+p).$$

Taking conditional expectation on both sides,

$$\mathcal{E}\{y(k+p\,|\,k)\} = \mathcal{E}\left\{\frac{F_p(q^{-1})}{C(q^{-1})}\right\} y(k\,|\,k) + \mathcal{E}\left\{\frac{F_p(q^{-1})B(q^{-1})}{C(q^{-1})}\, u(k+p-d\,|\,k)\right\} \\ + \mathcal{E}\{E_p(q^{-1})e(k+p\,|\,k)\}.$$

Since $E_p(q^{-1})$ is of degree p-1,

$$\mathcal{E}\{E_p(q^{-1})e(t+p)\} = 0;$$

again by definition $\mathcal{E}\{y(k+p|k)\} = \hat{y}(k+p|k)$, and since the inputs u(k+p-d), u(k+p-d-1), etc. are assumed to be known, the minimum mean square error prediction is given by

$$C(q^{-1})\hat{y}(k+p\,|\,k) = F_p(q^{-1})y(k) + E_p(q^{-1})B(q^{-1})u(k+p-d). \tag{5.2.3}$$

The prediction error is given by

$$\varepsilon(k+p\,|\,k) = y(k+p) - \hat{y}(k+p\,|\,k) = E_p(q^{-1})e(k+p).$$

So the prediction error will have the following statistical properties:

(i) Mean: $\mathcal{E}\{\varepsilon(k+p)\} = 0.$

(ii) Variance: $\mathcal{E}\{\varepsilon(k+p|k)^2\} = (1 + e_1^2 + e_2^2 + \ldots + e_{p-1}^2)\sigma^2,$

where

$$\mathcal{E}\{e^2(k+p)\,|\,y(k),\ y(k-1),\ldots\} = \sigma^2.$$

Summarizing the p-step explicit prediction procedure:

(1) Estimate the parameters of the model (5.2.1).

(2) For each p, compute parameters of $E_p(q^{-1})$ and $F_p(q^{-1})$ from the identity (5.2.2) (see Appendix 5A for a recursive algorithm and its implementation).

(3) Compute the optimal prediction $\hat{y}(k+p|k)$ using the estimated parameters from (5.2.3).

Remarks

In most practical cases, the noise in (5.2.1) may be due to sources external to the process or due to other uncertainties in the model. The correct estimation of the parameters of $C(q^{-1})$ is difficult unless the disturbances are frequent and have well defined statistical properties, which is rare in practice. Hence $C(q^{-1})$ may be assumed to be 1, implying $A(q^{-1})/C(q^{-1})$ and $B(q^{-1})/C(q^{-1})$ being absorbed into $A(q^{-1})$ and $B(q^{-1})$ respectively. Alternatively, a fixed noise observer polynomial $T(q^{-1})$ may be used to represent prior knowledge about the noise process, as explained in Sec.2.4.2. A typical choice for $T(q^{-1})$ is a first order polynomial, say $T(q^{-1}) = 1-0.7q^{-1}$. $T(q^{-1})$ replaces $C(q^{-1})$ in the identity (5.2.2), and the predictor (5.2.3) becomes

$$\hat{y}(k+p|k) = F_p(q^{-1})y_f(k) + E_p(q^{-1})B(q^{-1})u_f(k+p-d),$$

where $y_f(k) = y(k)/T(q^{-1})$, and $u_f(k) = u(k)/T(q^{-1})$.

ARIMAX model based design

Consider the process model

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k)/\Delta, \qquad \Delta = 1-q^{-1}.$$

Introduce the identity

$$C(q^{-1}) = E_p(q^{-1})\Delta A(q^{-1}) + q^{-p}F_p(q^{-1}).$$

Following the same procedure as for ARMAX model, the optimal p-step ahead prediction is given by

$$C(q^{-1})\hat{y}(k+p|k) = F_p(q^{-1})y(k) + E_p(q^{-1})B(q^{-1})\Delta u(k+p-d).$$

The prediction error is given by

$$\varepsilon(k+p|k) = E_p e(k+p).$$

5.2.2 Implicit (Direct) Prediction

ARMAX model based design

Consider the process model (5.2.1):

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k).$$

Following (5.2.3), and dropping the index (q^{-1}) for clarity,

$$\begin{aligned}\hat{y}(k|k-p) &= (1-C)\hat{y}(k|k-p) + Fy(k-p) + E_pBu(k-d)\\ &= \phi^T(k-p)\theta(k-p), \qquad (5.2.4)\end{aligned}$$

where

$$\begin{aligned}\phi^T(k-p) = [&-\hat{y}(k-1|k-p-1),\dots,-\hat{y}(k-n|k-p-n),\\ &y(k-p),\dots,y(k-p-n+1),\ u(k-d),\dots,\ u(k-d-p-n+1)],\end{aligned} \qquad (5.2.5)$$

θ being the associated parameter vector. The parameters of the predictor can be estimated from

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{k}(k)\varepsilon(k), \qquad (5.2.6)$$

where

$$\begin{aligned}\varepsilon(k) &= y(k) - \hat{y}(k|k-p)\\ &= y(k) - \phi^T(k-p)\hat{\theta}(k-1), \qquad (5.2.7)\end{aligned}$$

and $\mathbf{k}(k)$ is the Kalman gain of the estimator (as discussed in Sec.3.4.1).

Note that $\hat{y}(k|k-p)$ in (5.2.7) is computed using posterior parameter values $\hat{\theta}(k-1)$, and $\hat{y}(k-1|k-p-1)$ etc. in (5.2.5) can also be computed in the same way.

Summarizing the prediction procedure:

(i) Estimate predictor parameters $\theta(k)$ using (5.2.6).

(ii) Compute the p-step ahead prediction

$$\hat{y}(k+p|k) = \phi^T(k)\hat{\theta}(k). \qquad (5.2.8)$$

A comparative study of the explicit and the implicit methods of prediction is presented in Sec.5.3.4.

Self-tuning property

In both the explicit and the implicit approaches, the prediction problem is solved by considering a *separation* between the estimation of the parameters used in the predictor and computation of the predicted values. The

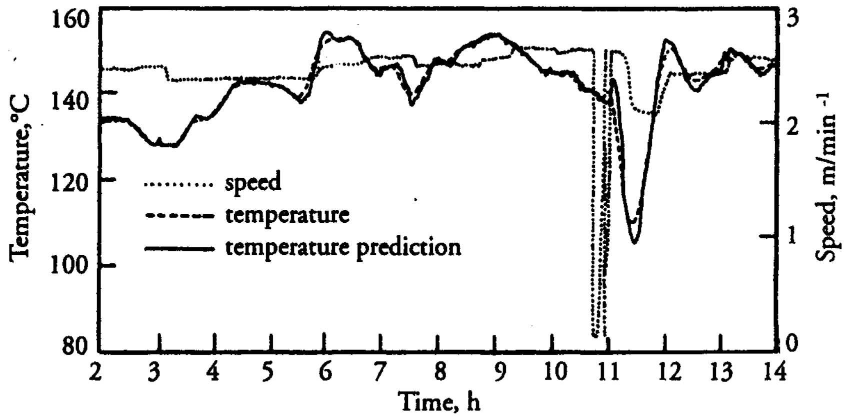


Figure 5.2.2 8-min ahead implicit prediction of temperature with discontinuity of service.

estimated parameters are used in the predictor, as if they were the true ones; this concept is known as *certainty equivalence*. Eventually, as the parameters converge to the true values, the predictor will also converge to the optimal minimum mean square error predictor; this is the self-tuning property of the predictor. The self-tuning property is further discussed in Sec.12.2.1.

Example 5.2.2 Prediction of waste gas temperature in the sintering process

A detailed description of the sintering process is given in Sec.5.6.1. The waste gas (WG) temperature being primarily related to the strand speed, the prediction problem concerns a one input (i.e. strand speed, u) and one output (i.e. WG temperature, y) process. The u and y data are available every 2-minutes. Consider using the implicit predictor (5.2.8).

Firstly, the time delay between a change in u and the consequent response in y is determined using historical data as 8 minutes (so d = 4). The predictor (5.2.4) is assumed to have two $E(q^{-1})$ and two $F(q^{-1})$ parameters. The mean and variance extracted data are used to estimate the parameters in (5.2.4) using the recursive least squares method. The 4-step ahead (i.e. 8 min ahead) prediction is produced; as shown in Fig.5.2.2.

In the present case, there was a brief discontinuity of service when the strand stopped moving towards the end of the 10th hour. As the speed signal (u) dropped to an abnormally low value, the parameters were frozen and the parameter estimation was suspended. The updating of the measurement vector continued. The predicted value of WG temperature was assumed to be the current value. After the operation was restored, the steady parameter values before stoppage were used for prediction until the newly estimated parameters resettled.

5.3 CONSTRAINED MEAN SQUARE ERROR PREDICTION

One of the ways of enriching the prediction algorithm with subjective knowledge is to use constrained minimum mean square error cost function, where in addition to the prediction error, the prediction increments are also costed. The consequent predictions are expected to be robust and meaningful.

5.3.1 Why Constrain Prediction

There are inherent constraints on the performance of most real-life processes. For example, the maximum or minimum values that the output can reach can be bounded, or say the maximum rate at which a variable can change with time may be limited etc. The prediction will be meaningful if such constraints of the process are not disregarded. Consider the following examples.

(a) If the process is inherently rate limited, there is sense in constraining the sequential increments in multistep prediction. For example, the silicon content in molten iron can change only within certain limits during a tapping operation of the blast furnace.

(b) Some processes show strong periodical links, which may justify constrained prediction. For example, in the case of power-load forecasting, the load during the following few hours of a particular day may not be very different from the load at the same time in the previous week; hence in computing the hourly predictions, it would be wise to penalize the difference of the load with that at the same time in the previous week. Again, additional constraints on sequential increments of the predicted load demand may also be imposed.

If the process is deterministic and the model is representative, the constraints inherent with the process may be built into the model. However for most cases, the model itself may not adequately incorporate the limitations of the process.

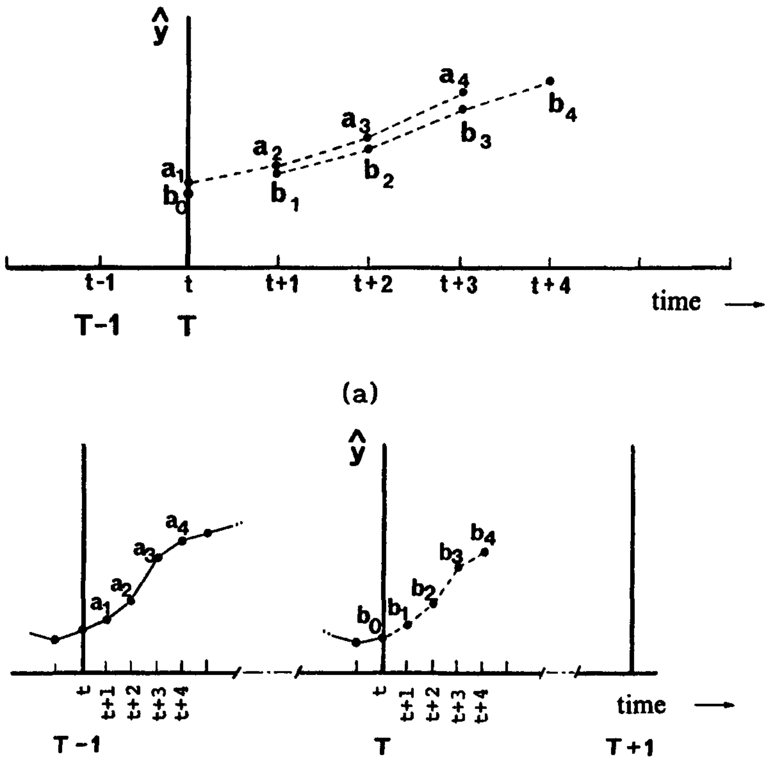
The minimum mean square error predictor expects all the characteristics of the process to be incorporated in the model. In addition to the process being correctly modelled, the measurements have to be noise free in order that sensible prediction can be produced. The additional costing on the prediction increments can protect a predictor from the influence of the noise associated with the data; at the same time, the user can incorporate specific subjective knowledge about the process in the predictor through the proper choice of the prediction increments to be costed, and thereby improve the quality of prediction.

5.3.2 Cost Criteria

In Fig.5.3.1(a), let a_1, a_2, a_3 and a_4 be the 1, 2, 3 and 4 -step ahead prediction respectively produced at time T-1, and b_1, b_2, b_3 and b_4 be similar predictions produced at time T. Here T, T-1,... etc. are the prediction intervals, and k, k+1,... etc. are the sampling intervals or the time steps over which predictions are performed. In the case of a periodic process (Fig.5.3.1(b)), T, T-1, may relate to consecutive periods where a_1, a_2, a_3 and a_4 are the corresponding actual outputs pertaining to the last period T-1. In both the cases, the predictions, b_1, b_2, b_3 and b_4 may be evaluated at time T with constraint imposed on the prediction increments as follows:

(a) J_{c1} costing where delayed increments between a_1 and b_1, a_2 and b_2, a_3 and b_3, and a_4 and b_4 etc. are penalized.

For example, in the case of electric power load on a substation (see Example 7.5.2) a_1, a_2, a_3 and a_4 may be the



(a)

(b)

Figure 5.3.1 (a) Typical multistep prediction of a time series (b) Typical multi-step prediction for a periodic process.

hourly load measured at the same time during a previous week, where b_1, b_2, b_3 and b_4 are the hourly predictions being produced in the present week. Since the power consumption for a particular day of the week is expected to follow a certain pattern, it is wise to cost the differences between the hourly predictions of consecutive weeks.

(b) J_{c2} costing where sequential increments between b_o and b_1, b_1 and b_2, b_2 and b_3, b_3 and b_4 etc. are penalized, b_o being the present measurement.

For example, in the prediction of the temperature of a soaking pit (discussed in Sec.6.8) the change in the temperature of the pit over a certain length of time (say ten minutes) is expected to be within certain limits (say within ± 20°C), because of the high time-constant of the process. Hence the sequential increments may be costed accordingly.

(c) J_{c3} costing where positional differences between a_2 and b_1, a_3 and b_2, a_4 and b_3 etc. are penalized.

For example, consider the case of the prediction of the quality of sinter (Sec.5.6), where hourly predictions are produced. In evaluating the prediction for a particular time instant in the future, it makes sense to take into account the prediction made at the preceding hour for the same instant, since the process characteristics are not expected to change by a great deal in one hour.

It is clear that the prediction increments that should be constrained and the time increments that should be considered are dependent on the nature of the process. Consider a generalized cost criterion

$$J_c = J_m + J_{inc}, \tag{5.3.1}$$

where by definition,

$$J_m = E\{(\varepsilon(k+p))^2\}, \qquad \varepsilon(t+p) = y(k+p) - \hat{y}(k+p|k),$$

and J_{inc} is the cumulative costing on (say n) different types of prediction increments:

$$J_{inc} = E\{(\varepsilon_{inc})^2\} = E\{\sum_{i=1}^{n}\lambda_i(\varepsilon_i)^2\}. \tag{5.3.2}$$

λ_i is a scalar constant. Thus through J_{inc}, the cost function J_c permits simultaneous penalization of different types of prediction increments ε_{inc}.

If only delayed prediction increments for a periodic type process are to be penalized (as discussed in (a) above), the cost becomes

$$J_{c1} = E\{(\varepsilon(k+p))^2 + \lambda_1\varepsilon_1^2\},$$

where

$$\varepsilon_1 = \hat{y}(k+p|k) - \hat{y}(k+p|T-1), \tag{5.3.3}$$

$\hat{y}(k+p|k)$ being the same as $\hat{y}(k+p|T)$.

If only sequential prediction increments are to be penalized as discussed in (b) above, the cost becomes

$$J_{c2} = E\{(\varepsilon(k+p))^2 + \lambda_2\varepsilon_2^2\},$$

where

$$\varepsilon_2 = \hat{y}(k+p|k) - \hat{y}(k+p-1|k). \tag{5.3.4}$$

If only positional differences in prediction are to be costed, as discussed in (c) above, the cost becomes

$$J_{c3} = E\{(\varepsilon(k+p))^2 + \lambda_3\varepsilon_3^2\},$$

where

$$\varepsilon_3 = \hat{y}(k+p|k) - \hat{y}(k+p|k-1). \tag{5.3.5}$$

Thus ε_{inc} in (5.3.2) can be appropriately defined by the designer, who can also consider a combination of different types of prediction increments which will additively constitute ε_{inc}.

Minimization of the cost (5.3.1) leads to

$$\frac{\partial J_c}{\partial\hat{y}(k+p|k)} = -2(y(k+p) - \hat{y}(k+p|k)) + 2\sum_{i=1}^{n}\lambda_i(\varepsilon_i)$$
$$= 0;$$

or

$$\varepsilon(k+p) = \sum_{i=1}^{n}\lambda_i(\varepsilon_i). \tag{5.3.6}$$

Remarks

The different cost criteria offer different features and are applicable to different types of processes; hence a general comparison is not possible. However, the following features may be noted. In the case of the costing J_{c1} and J_{c3}, the prediction inaccuracy due to a sudden disturbance would propagate through the prediction intervals (for example T-1 to T), whereas in the case of the costing J_{c2}, where sequential increments are costed, the effect of a similar disturbance would be rejected in one prediction interval. In all the three cases the effect of disturbance would be reduced.

5.3.3 Prediction Formulations

Explicit approach

Consider the ARIMAX model

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k)/\Delta, \tag{5.3.7}$$

where $C(q^{-1})$ is a stable polynomial; it is assumed that the known input u, the output y and the noise are mean square bounded.

Since the correct estimation of the parameters of $C(q^{-1})$ is difficult, let a (known) noise observer polynomial $T(q^{-1})$ be used. Define the identity

$$T(q^{-1}) = E_p(q^{-1})\Delta A(q^{-1}) + q^{-p}F_p(q^{-1}), \tag{5.3.8}$$

where

$$E_p(q^{-1}) = 1 + e_1q^{-1} + \ldots + e_{p-1}q^{-p+1}, \quad \text{and}$$
$$F_p(q^{-1}) = f_o + f_1q^{-1}+ \ldots + f_nq^{-n};$$

the degree of F_p, $\delta E_p = p-1$, and $\delta F_p<\delta(\Delta A)$.

From (5.3.7) and (5.3.8), omitting the symbol (q^{-1}) for simplicity,

$$y(k+p) = F_py(k)/T + E_pB\Delta u(k+p-d)/T + E_pe(k+p); \tag{5.3.9}$$

since E_p is of degree p-1, $E_pe(k+p)$ is independent of the rest of the terms. Hence the constrained minimum mean square error prediction is given by

$$\hat{y}(k+p|k) + \varepsilon(k+p)$$
$$= F_py(k)/T + E_pB\Delta u(k+p-d)/T \tag{5.3.10}$$

where $\varepsilon(k+p)$ is given by (5.3.6).

Summarizing the constrained explicit prediction procedure:

(1) Estimate the process parameters in

$$A(q^{-1})\Delta y_f(k) = B(q^{-1})\Delta u_f(k-d) + e(k), \quad (y_f = y/T)$$

using the recursive least squares (RLS) method.

(2) For each value of p, determine the parameters $E_p(q^{-1})$ and $F_p(q^{-1})$ from the identity (5.3.8).

(3) Define the constraint on the prediction increments (5.3.6).

(4) Compute the p-step prediction $\hat{y}(k+p|k)$ using (5.3.10).

Implicit approach

The parameters of the predictor in (5.3.9) can be directly estimated from

$$y(k) = F_py(k-p)/T + G_p\Delta u(k-d)/T + E_pe(k), \tag{5.3.11}$$

where $G_p = E_pB$, $\delta E_p = p-1$, $\delta G_p = \delta E_p+\delta B = n+p-1$, $\delta F_p = \delta A$.

Note that the data in (5.3.11) are not properly balanced, as the output data are positional or actual, whereas the input data are incremental; this is likely to affect the quality of estimation. Hence introduce the identity

$$F_p(q^{-1}) = T(q^{-1}) + \Delta F_p'(q^{-1}). \tag{5.3.12}$$

So the estimator (5.3.11) may now be expressed as

$$y(k) - y(k-p) = F_p'\Delta y_f(k-p) + G_p\Delta u_f(k-d) + E_pe(k), \tag{5.3.13}$$

where $y_f = y/T(q^{-1})$ and $u_f = u/T(q^{-1})$ and the degree of F_p', $\delta F_p' = \delta F_p - 1$.

When the prediction horizon p is greater than the time delay d, the noise term in (5.3.13) is no longer uncorrelated with the data; this can be avoided in order to use the RLS estimation method, by modifying the time indexing as follows:

$$\begin{aligned} &y(k+p-N) - y(k-N) \\ &\quad = F_p'\Delta y_f(k-N) + G_p\Delta u_f(k+p-d-N) + E_pe(k+p-N), \end{aligned} \tag{5.3.14}$$

where N is the maximum length of the prediction horizon.

The p-step ahead implicit predictor is given by

$$\begin{aligned} &\hat{y}(k+p\,|\,k) + \varepsilon(k+p) \\ &\quad = y(k) + F_p'\Delta y_f(k) + G_p\Delta u_f(k-d), \end{aligned} \tag{5.3.15}$$

where $\varepsilon(k+p)$ is given by (5.3.6).

Example 5.3.3 Explicit prediction with J_{c1} costing

A simulated process, controlled by an LQG controller (discussed in Sec.13.6) is considered (note that the prediction procedures are independent of the control policy, if any, in use). The future values for the control input are assumed to remain unchanged at the latest known value. An RLS estimator with a forgetting factor of 0.99 is used and no prior knowledge of plant parameters is assumed.

Consider the model

$$(1-1.7q^{-1}+0.7q^{-2})y(k) = (1-1.5q^{-1})u(k-1) + (1-0.6q^{-1})e(k),$$

with RMS value of noise equal to 0.1. The process is also being acted on by step load disturbances in the output (e_y)

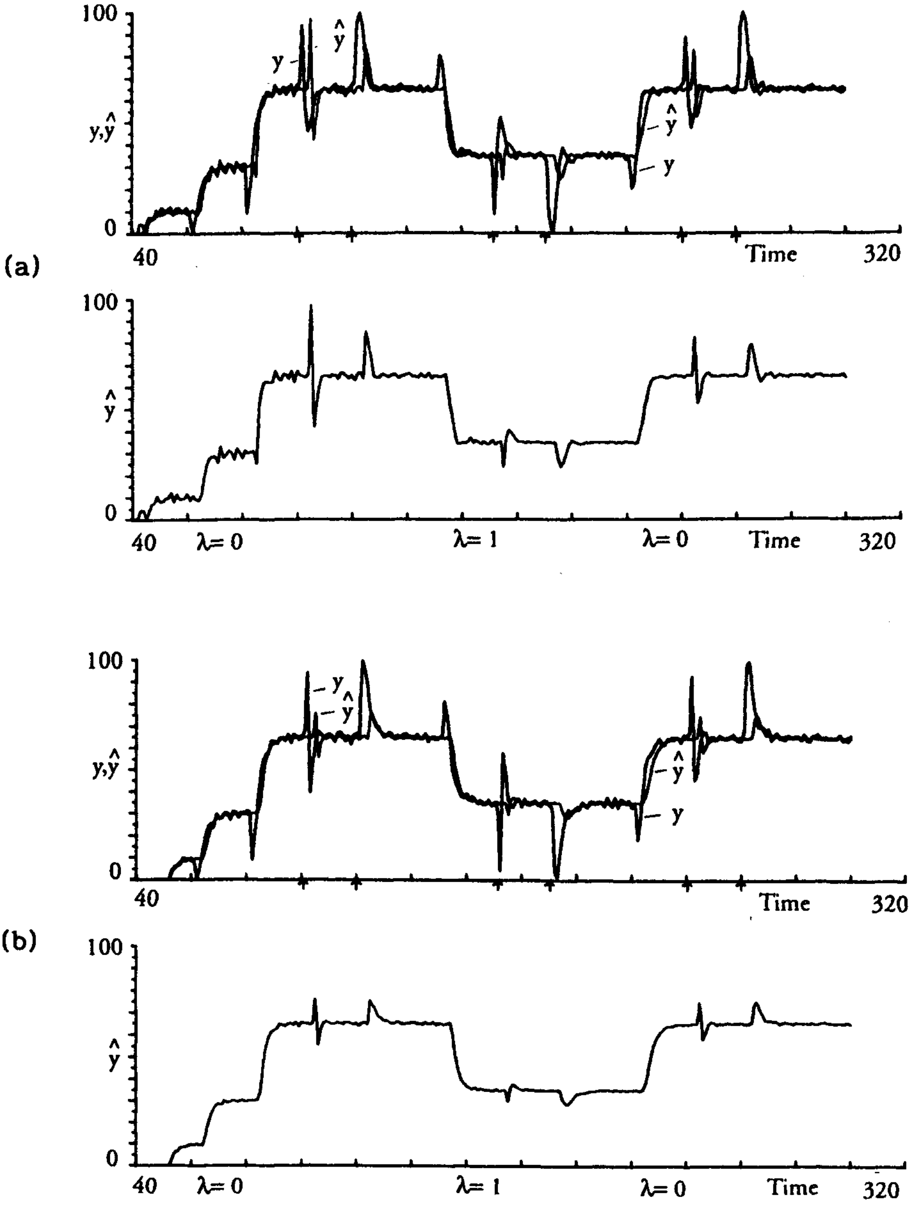


Figure 5.3.2 (a) Constrained explicit prediction with λ, without T-filter.
(b) Constrained explicit prediction with λ and with T-filter. (——— actual output, ——— prediction).

as well as in the input (e_u). In this example, first an e_y and an e_u of magnitude +3 are applied. Following this, λ_2 costing of magnitude 1 is introduced on the prediction increment, and the same disturbances e_y and e_u are re-applied in the reverse direction. Finally, the cost on the prediction-increment is withdrawn, and e_y and e_u of the same magnitude are re-applied. The exercise is repeated with an assumed noise observer filter, $T(q^{-1}) = (1-0.5q^{-1})/0.5$.

The results are shown in Figs.5.3.2a and 5.3.2b. It can be seen that the costing λ_2 is very effective in reducing the effects of the disturbances e_y and e_u on the predictor. The results also show that disturbance rejection performance of the predictor is much improved when noise observer polynomial T is used; this protects the estimator and also since the disturbances are filtered, they have milder effects on the predictor.

5.3.4 Comparative Study

The explicit or indirect method of prediction requires only one stage of parameter estimation, followed by solution of the identity (5.3.8) for each value of prediction step p. Since this identity can easily be solved recursively (Appendix 5A), the computational load is not much. However the accuracy of prediction mainly depends on one-stage estimation.

On the other hand, the implicit or direct method of prediction uses the same formulation for estimation and prediction, which can make the predictor well behaved but the computational expense is higher as it is necessary to run p number of estimators in parallel. One alternative to multistage estimation is to use the same covariance matrix for the part of the data vector in (5.3.15) which remains unchanged as the prediction horizon increases as explained below.

In (5.3.14) the degrees $\delta F_p'$ (= δA-2), and δG_p (= n+p-1), i.e. the number of parameters in G_p increases with p. If for every p>1, the 1st p parameters of G_p are assumed to remain unchanged (at the values obtained from the estimation of the previous step, p-1). The data vector on the right-hand side of (5.3.14) remains unchanged when arranged as in (5.3.16). So the N stage estimation problem is configured into a one-stage estimation (pertaining to p=1) and (N-1) stages of parameter reconstruction problem (pertaining to p≥2):

$$y(k+p-N) - y(k-N) - \hat{G}_1\Delta u_f(k+p-k-N)$$
$$= F_p'\Delta y_f(k-N) + G_2\Delta u_f(k+p-k-N) + E_p e(k+p-N), \qquad (5.3.16)$$

where

$$G_2(q^{-1}) = G_p(q^{-1}) - G_1(q^{-1}), \qquad p \geq 2,$$
$$G_1(q^{-1}) = g_o + g_1 q^{-1} + \ldots + g_{p-2} q^{-p+2}.$$

$\hat{G}_1$, the parameters of the first (p-1) terms of $G_p(q^{-1})$, are assumed to have been obtained from the earlier stages of estimation and only the parameters of $F_p'(q^{-1})$ and $G_2(q^{-1})$ are determined. Note that the noise terms in (5.3.16) are uncorrelated with the data.

The estimation procedure can be summarized as follows:

(1) Estimate the parameters of (5.3.16) for p=1 when G_1=0.

(2) For the next value of p, the 1st parameter of the last G_2 is fixed and goes to G_1 and the parameters (θ) of F′ and the new G_2 are determined from

$$\hat{\theta}_{p+1} = \hat{\theta}_p + \mathbf{k}(1)\mathrm{x}(\text{Estimation error})$$

where only the estimation error changes from one step to the next; **k**(1) is the fixed Kalman estimator gain vector obtained for p = 1 from (1).

(3) Repeat (2), until p = N.

5.4 MULTISTEP PREDICTION THROUGH PROCESS MODEL RECURSION

If the prediction horizon p is greater than the degree of the noise process, the prediction $\hat{y}(k+p|k)$ can be obtained by iterating the process model as explained below.

For the sake of simplicity, without any loss of generality, consider the ARMA representation of the process:

$$A(q^{-1})y(k) = C(q^{-1})e(k), \qquad (5.4.1)$$

where A and C are stable polynomials of degree n and m respectively in the backward shift operator q^{-1}. Hence

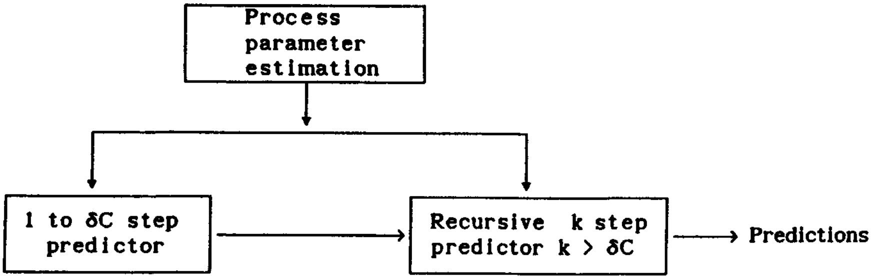

Process parameter estimation
1 to δC step predictor
Recursive k step predictor k > δC
Predictions

Figure 5.4.1 Schematic diagram of multistep predictor through process model recursions.

$$y(k+i|k) + a_1y(k+i-1|k)+\ldots+ a_ny(k+i-n|k)$$
$$= e(k+i|k) + c_1e(k+i-1|k) +\ldots+ c_me(k+i-m|k), \qquad (5.4.2)$$

where $y(k+i|k) = y(k+i)$ for $i = 0,-1,\ldots$etc., and $e(k+i|k) = 0$ for $i = 1, 2,\ldots$ etc. For $i \geq m+1$, the right-hand-side of (5.4.2) vanishes, leading to

$$y(k+i|k) = -a_1y(k+i-1|k)-a_2y(k+i-2|k)-\ldots-a_ny(k+i-n|k). \qquad (5.4.3)$$

Hence, the i-step ahead prediction $\hat{y}(k+i|k)$ in (5.4.3) will be optimal, if

(1) the parameters a_i are true,
(2) $i>\delta C$, and
(3) $\hat{y}(k+i-1|k)$, $\hat{y}(k+i-2|k)$, ..., etc. are optimal.

If an ARIMAX model is considered for the process, for $>\delta C$, the predictor is given by

$$\tilde{A}(q^{-1})\hat{y}(k+p|k) = B(q^{-1})\Delta u(k+p-d), \qquad (5.4.4)$$

where

$$\tilde{A} = (1-q^{-1})A \quad = 1 + \tilde{a}_1q^{-1} + \tilde{a}_2q^{-2} + \ldots+ \tilde{a}_{n+1}q^{-n-1},$$

$$\tilde{A}(q^{-1})\hat{y}(k+p|k) \quad = \hat{y}(k+p|k) + \tilde{a}_1\hat{y}(k+p-1|k) + \tilde{a}_2\hat{y}(k+p-2|k) + \ldots + \tilde{a}_{n+1}\hat{y}(k+p-n-1|k).$$

Summarizing the prediction procedure (Fig. 5.4.1):

(1) Estimate the parameters of the process model (5.4.1).
(2) Compute the predictions $\hat{y}(k+p|k)$ for $p = 1$ to δC, using

any suitable method; δC is the degree of the noise process.

(3) Compute the prediction $\hat{y}(k+p|k)$, for $p>\delta C$, by iterating the process model recursively as in (5.4.3) for ARMA or as in (5.4.4) for ARIMAX process.

5.5 CASE STUDY: PREDICTION OF PRODUCT QUALITY IN IRON-ORE SINTERING PROCESS

A multivariable predictor is used to predict product quality in an iron-ore sintering process. The problem is configured as a two input two output prediction problem. Since the data on the product quality are obtained from off-line measurements, real-time filtering is used before the data are used for prediction.

5.5.1 Process Description and Prediction Problem

Iron-ore sinter is a preprocessed feed material for blast furnaces. A schematic diagram of the sintering process is shown in Fig.5.5.1. Iron bearing fine materials are mixed with coke breeze, flux, water etc. to form the raw mix. The material is loaded onto a moving strand and is levelled to form a flat bed, the surface of which is ignited as it passes under an ignition hood. As the bed travels horizontally, a combustion zone is drawn downwards through the material under the influence of an exhaust fan, thereby driving off the volatiles and fusing the material to form sinter. Usually, the speed of the strand is adjusted such that by the time the processed material arrives at the end of the strand, the combustion zone reaches the bottom of the bed and thus allows just sufficient on-strand time. At the other end of the strand, the processed material is unloaded, crushed and screened. The sinter fines (typically <5 mm) are returned to the mixing station for reprocessing and the sinter product is passed for use in the blast furnace.

The physical strength and the degree of oxidation, indicated by the FeO content, are the two important measures of quality of the sinter product. These measurements being available from infrequent off-line analyses, their prediction can be useful for the plant operators. The variables

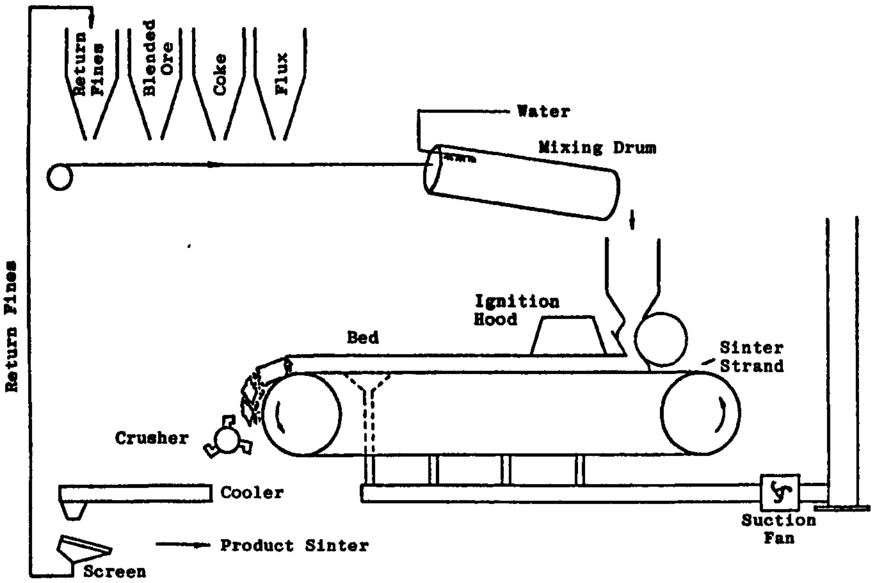


Figure 5.5.1 Schematic diagram of the Sintering process.

which most strongly influence the physical strength and the FeO content of sinter are the coke rate and the on-strand sintering time. If the on-strand process is strictly under control, the on-strand sintering time can be directly calculated from the strand speed; otherwise, both the strand speed and the waste gas (WG) temperature must be considered.

The present study is based on data collected from a sintering plant of rated capacity of 13000 tons per day. Average values of the strand speed, waste gas temperature and coke rate measurements were available at 15-min intervals. A temperature/speed factor, an auxiliary variable is used which is empirically computed as

$$\text{Temperature/speed factor}(k) = \text{Speed}(k) \times \frac{\text{WG temperature}(k) + \text{WG temperature}(k-1/2)}{2},$$

where one time step is considered to be 1 hour. The strength and FeO measurements were available alternately, every hour. The objective was to produce hourly predictions of strength and FeO based on the two inputs, coke rate and temperature/speed factor. From the knowledge of the process, the time delay associated with the coke rate was assumed to be 4 hours and that for the temperature/speed factor was assumed to be 3 hours.

Remarks: Measurements of FeO and sinter strength

FeO content is determined by an analytical method. A representative sample of the sinter product is prepared and is tested typically using an X-ray diffractometer.

The sinter strength is determined empirically. In a typical set up, 15 kg of representative sample is separated into a drum. The drum is rotated 200 times in 7.5 minutes, after which the product is treated in a vibrator for 3 minutes. The resulting +6.3 mm percentage of the product is expressed as the ISO strength index.

5.5.2 Data Preparation

The strength and the FeO measurement data required low-pass filtering because of (i) inhomogeneity of the solid samples that are tested and (ii) the empirical nature of the strength measurement. The other data being averaged, no filtering was necessary. Unfortunately low-pass filtering tends to produce time lag which deteriorates the real-timeliness of the data. Hence in the present study, an extended first-order Butterworth filter (Appendix 14A) was used. When a new measurement is available, the predictor is used to compute a one-step ahead prediction, and this predicted value is entered into the filter instead of the actual value.

The improvement in the performance may be explained as follows. A first-order Butterworth filter has a pole on the positive axis inside the unit disc, and a zero at -1. Ideally, the extended configuration introduces a zero at the origin in order to reduce the phase or time lag without altering the gain. In practice, due to the inaccuracy of one-step prediction, this zero tends to move towards the left of the origin and the reduction in time lag is less. Note that the bidirectional filtering, which is an off-line method, produces no lag, whereas the ordinary low-pass filtering produces significant lag. The present low-pass filtering method being a real-time approach is an ideal compromise. In the present case the ratio of (cut-off frequency)/(sampling frequency) is conservatively chosen as

0.15 such that the information content in the data is not affected. Real-time filtering is discussed in Sec.14.3.2.

5.5.3 Prediction Exercise

Problem formulation

Using the implicit prediction policy, the predictor is expressed as

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \phi^T \begin{bmatrix} \theta_1(k) \\ \theta_2(k) \end{bmatrix} + \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix},$$

where y_1 and y_2 represent the strength index and the FeO content respectively, e_i are the uncorrelated equation errors, ϕ and θ_i are the data and predictor parameter vectors respectively. For one-step ahead prediction:

$$\phi(k) = [y_1(k-1)\ y_2(k-1)\ y_1(k-2)\ y_2(k-2)$$
$$u_1(k-4)\ u_2(k-3)\ u_1(k-5)\ u_2(k-4)\ u_1(k-6)\ u_2(k-5)]^T,$$

where u_1 and u_2 are coke rate and temperature/speed factor respectively. Thus the multivariable prediction may be solved as a multi-input/single output problem:

(i) Parameter estimation:

$$y_i(k) = \phi^T(k)\theta_i(k) + e_i(k), \quad i = 1, \text{ and } 2;$$

(ii) p-step ahead prediction:

$$\hat{y}_i(k+p|k) = \phi^T(k+p)\hat{\theta}_i(k).$$

In the present application, the strength and the FeO measurements were available alternatively, every hour (i.e. measurements of strength or FeO are available at two-hour intervals). So when a new strength measurement is available, first the multivariable predictor is used to compute one-step ahead prediction then the extended low-pass filtering is performed, and the filtered value is used to evaluate multistep prediction. For the prediction of the sinter strength, the predictor uses information of the present and past coke rates, present and past temperature/speed factors,

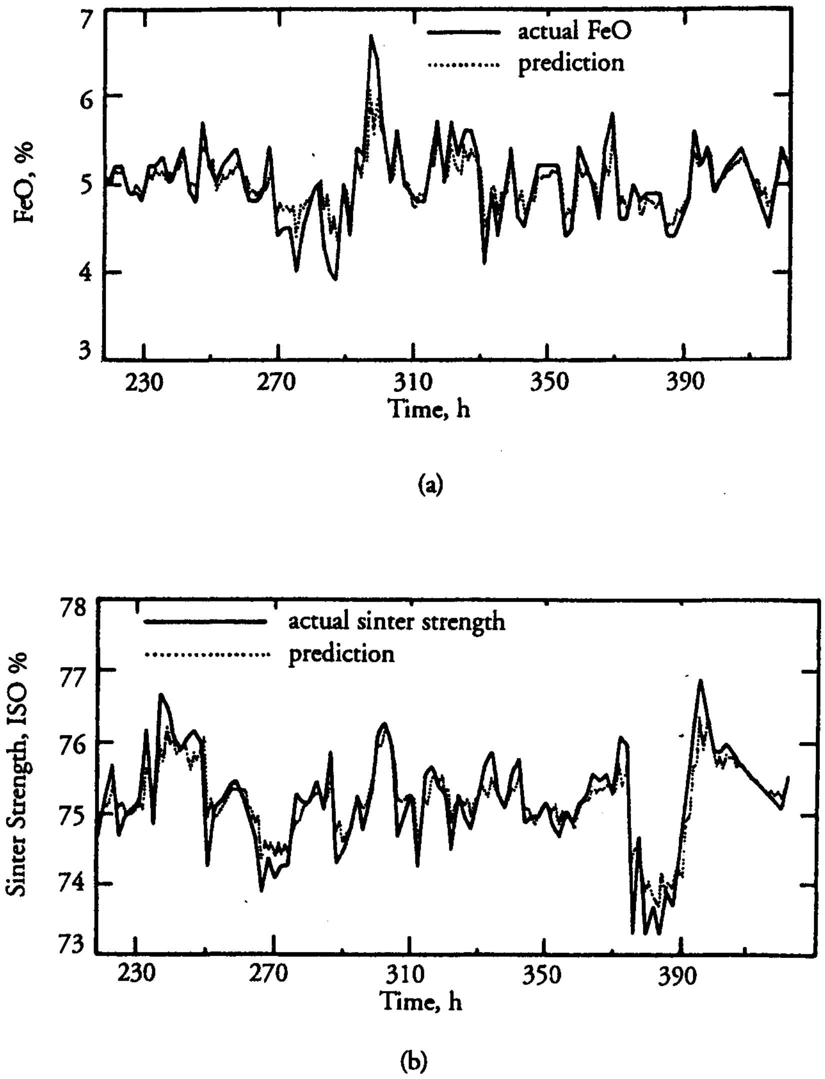


Figure 5.5.2 (a) One to two-hour ahead prediction of FeO content of sinter product,
(b) one to two-hour ahead prediction of sinter strength.

present and past strength measurements, measurement and prediction of FeO content produced in the past hour. In the following hour, as a new FeO measurement is obtained, prediction of FeO content is computed the same way. .

Results and discussions

In the present application, the predictor parameter estimator is allowed to free-wheel through the hourly interpolated data sets between the two-hour intervals when the strength and FeO measurements are available. This procedure is found to produce better consistency of the parameters than when the estimator operates at two-hour intervals alone. The predictions of FeO and sinter strength are run in parallel, and the results of one are used in predicting the other. The results of one to two-hour predictions are presented in Fig.5.5.2(a) and Fig.5.5.2(b) (i.e. after every two hours, one-hourly and two-hourly predictions are produced). Thus in the present case, four recursive least squares estimators are run in parallel. Note that although the process is complex, the predictions produced are quite close. It is found that the two-hourly predictions are marginally inferior, if the prediction is computed by recursion of the process model instead of using the optimal prediction procedure.

5.6 CONCLUSIONS

Sensible prediction demands careful design of both the estimator and the predictor. In the present context, measures such as use of incremental data in an ARIMAX framework, use of noise observer filter, provision ensuring sensible steady-state values for the estimator covariances etc. can help the process of estimation. It is argued that real-time prefiltering of the data with minimum loss of phase information or real-timeliness, can also improve the quality of data for estimation and prediction.

It is emphasized that the predictor need not be a reformulation of the estimator, like the conventional minimum variance predictor. Improved prediction and better disturbance rejection is possible through constrained minimum variance strategy, where prediction increments or change of predicted levels over specific time intervals are constrained. This approach permits incorporation of subjective information about the process into the predictor, which can go a long way in improving the information content in the predicted values.

REFERENCES

Remarks: Adaptive minimum mean square error prediction and the self-tuning concepts are introduced in [15], and are further extended in [3,7]. The analysis of ARMA processes in Markovian framework used in Sec.5.4 is studied in [1]. The concept of constrained prediction is introduced in [9]. The multivariable design of the predictor is presented in [14], and is also discussed in Appendix 5B. The use of noise observer polynomials features in [2, 13]. The prediction applications are discussed in [4,8,10,12]. A review of k-step ahead predictors appear in [5]. Adaptive prediction algorithms and their properties are discussed in the text [6].

[1] Akaike, H. (1974): 'Markovian representation of stochastic processes and its applications to analysis of autoregressive moving average processes', *Annals of the Institute of Statistical Mathematics*, 26, 363-387.

[2] Clarke, D.W., C. Mohtadi, and P.S. Tuffs (1987): 'Generalized predictive control - Part II, extensions and interpretations', *Automatica*, 23(2), 149-160.

[3] De Keyser, R.M.C., and A.R. Van Cauwenberghe (1982): 'Simple self-tuning multistep predictions', in G.A. Bekey and G.N. Saridis (eds), *Identification and System Parameter Estimation*, 6th IFAC Symp., Arlington, MacGregor and Werner, Washington, 1558-1563.

[4] De Keyser, R.M.C., and A.R. Van Cauwenberghe (1981): 'A self-tuning multistep predictor application' *Automatica*, 17(1), 167-174.

[5] Favier, G., D. Dubois, and C. Rougerie (1988): 'A review of k-step ahead predictors', *IFAC Proc. Systems Ident. and Param. Est.*, Vol.2 (Preprints), 887-893.

[6] Goodwin, G.C., and K.S. Sin (1984): *Adaptive Filtering Predicton and Control*, Prentice-Hall, Englewood Cliffs.

[7] Holst, J. (1977): 'Adaptive prediction and recursive parameter estimation', Report No. TFRT-1013-1-206, Lund Inst. of Tech., Sweden.

[8] Kanjilal, P.P., and E. Rose (1986): 'Application of multivariable self-tuning predictor', *Proc. IEEE*, 74(2), 377-379.

[9] Kanjilal, P.P., and D.W. Clarke (1987): 'Prediction of stochastic processes using self-tuning principles', *Int. J. Systems Sci.*, 18(2), 371-388.

[10] Kanjilal, P.P., and E. Rose (1986): 'Application of adaptive prediction and control methods for improved operations of the sintering process', *Ironmaking and Steelmaking*, 13(6), 289-293.

[11] Ljung, L., and T. Söderström (1983): *Theory and Practice of Recursive Identification*, MIT Press, Cambridge.

[12] Sadiq, A., and A.K. Sinha (1986): 'Self-tuning predictions with application to river flow prediction', *Int. J. Systems Sci.*, 17(4), 655-668.

[13] Soeterboek, R. (1992): *Predictive Control: A Unified Approach*, Prentice Hall, New York.

[14] Tanttu, J.T. (1980): 'A self-tuning predictor for a class of multivariable stochastic processes', *Int. J. Control*, 32, (2), 359-370.

[15] Wittenmark, B. (1974): 'A self-tuning predictor', *IEEE Trans. on Automatic Control*, AC-19 (6), 848-851.

CHAPTER 6

KALMAN FILTER AND STATE-SPACE APPROACHES

Systems can be modelled using state variable representation. Kalman filter offers a method for state estimation and prediction.

6.1 INTRODUCTION

The system representations used so far have been based on input and output variables, which were assumed to be known quantities. Systems can also be modelled using state variables, which may or may not be measurable. Thus state-space representation permits incorporation of variables which can relate to internal behavior of systems that cannot be accessed or measured, along with the measurable external variables. Besides, the state-space approach also enables concise system representation which can be easily implemented using computers. This chapter is devoted to the state-space approach to modelling of systems or processes, and estimation and prediction of state variables.

The state-space model can be developed based on the input-output relationships or the prior knowledge of the physical laws governing the system. The modelling requires specification of the state variables, along with the input and output variables, and knowledge of the statistical nature of the uncertainties present. Once the model is developed, one of the main concerns is to produce estimation of the states. In state-space modelling, the term *estimation* is used in two different contexts, namely (i) estimation of the parameters (if unknown) of the state-space model, and (ii) estimation of the states, when the measurements are contaminated with additive noise. The parameter estimation problem has been discussed in detail in Chapter 3. A major part of this chapter is devoted to the problem of the estimation of the states using the Kalman filter.

The Kalman filter (Kalman, 1960) can produce optimal estimation of the states, when the additive measurement noise is Gaussian and independent of the measurements. One of the strongest features of Kalman's algorithm is that recursive implementation is possible. This added to the fact that state-space approach itself offers many structural and computational advantages, makes estimation of states using the Kalman filter an attractive proposition.

This chapter starts with the basics of state-space representation in Sec.6.2. The methods of state-space modelling are discussed next in Sec.6.3; both processes with or without noise are considered. Sec.6.4 presents modelling of periodic processes; state-space models are produced for the different structural components of periodic processes and prediction methods are developed. Sec.6.5 introduces the problem of optimal estimation of the states. Sec.6.6 is devoted to the study of the Kalman filter; the estimation algorithm is presented and its characteristic features are discussed. The prediction of the states, discussed in Sec.6.7, follows naturally from the state estimation. An application study relating to the estimation and prediction of the temperature in the soaking pit process at the finishing stages of steel-making is discussed in Sec.6.8; this study elaborates the issues involved in a complex industrial application, and the way the optimal estimation and prediction can be performed.

6.2 STATE-SPACE REPRESENTATION

A system is usually described by an *external model* expressed in terms of only the output of the system (for example, the time series model of the economic inflation of a country), or both the input and the output of the system (for example, the ARMA model of a furnace with fuel-gas flow into the furnace as input and the furnace temperature as output). The process variables of these models are directly measurable, and have physical connotations with respect to the system. However the external model provides no insight about the internal dynamics of the system.

The *internal model* or a state-space model offers an alternative approach to system representation. These models are expressed in terms of the outputs, the inputs and the states of system (Fig.6.2.1). The states contain complete historical information about the system, although the states may not have any physical meaning and they may not be directly measurable. The state-space model provides a

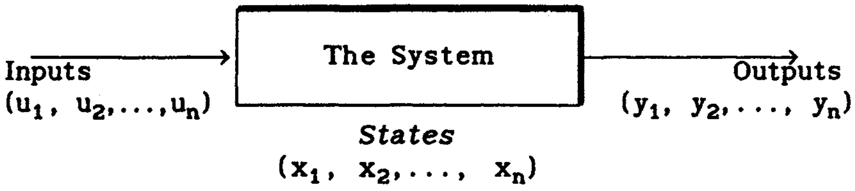

Inputs
$(u_1, u_2, \ldots, u_n)$
The System
Outputs
$(y_1, y_2, \ldots, y_n)$
States
$(x_1, x_2, \ldots, x_n)$

Figure 6.2.1 State-space representation schematic.

description of the internal and the external characteristics of the system.

A fundamental property of a dynamical system is that it has a *memory*. In other words, behaviour of the system at any instant depends on the variables currently acting on it, as well as on the variables that had acted on it in the past, information on which remains stored in the states of the system. Thus only the present state, and the present and future inputs are required to predict the future behaviour of the system. This characteristic of the state-space models of isolating the future from the past by incorporating all the past information in the current states is called the *Markov property* (see Sec.2.2.1). A stochastic process is called a Markov process, if

$$P\{x(k+1)\,|\,x(k),\ x(k-1),\ x(k-2),\ldots,\ x(0)\}$$
$$= P\{x(k+1)\,|\,x(k)\},$$

that is the conditional probability density function for x(k+1) depends only on its present value x(k) and not on any value in the past. Thus, the Markov property implies probabilistic causality.

State equations can be formulated in different forms to represent different types of processes. Unlike the inputs and the outputs, the state variables are not unique and the designer is free to attribute appropriate definitions to the states and formulate the model accordingly. The state of a system is a vector, and the state-space models are described by matrix operators.

Basic requirements for modelling

(a) To form a state-space model at least three variables are necessary: the input (u or e), the output (y) and the state (x) variables. The dimension of the state vector is at least equal to the *order* of the system.

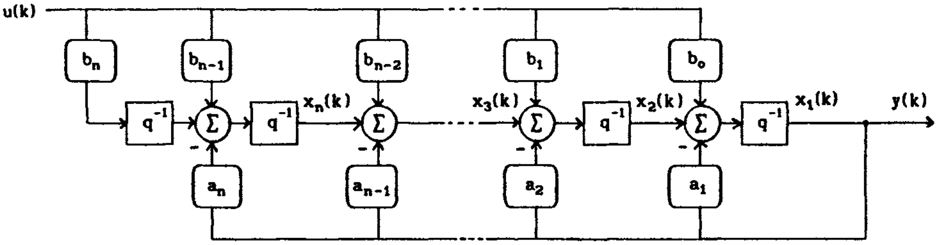


Figure 6.3.1 Structure of the process given by (6.3.1)

(b) The relationship between the inputs, the outputs and the state variables, through the system parameters has to be defined, and it may not be time invariant.
(c) The law for the transformation of the state vector from one time instant to the next should be stated; the initial value for the state $x(0)$ should also be known.
(d) The joint statistics of all variables should be known.

Examples of state-space models follow in the next section.

6.3 STATE EQUATIONS FROM DIFFERENCE EQUATION MODELS

The two broad classes of processes are those with or without the measurement noise in the output.

6.3.1 Processes without Measurement Noise

Consider the difference equation model

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n)$$
$$= b_ou(k-1) + b_1u(k-2) + \dots + b_nu(k-n-1), \qquad (6.3.1)$$

where y is the output and u is the input. There may be many state-space realizations of (6.3.1), each comprising basic structural blocks like adders, scalars, delay or backward shift operators etc. A typical representation of (6.3.1) is shown in Fig.6.3.1.

Choosing the state variables x_i as the outputs of the shift operators, the following state relations are obtained.

Let the state variables be defined as follows:

$$\begin{aligned}
x_1(k) &= y(k), \text{ and since } q^{-1}x_1(k+1) = x_1(k),\\
x_1(k+1) &= -a_1y(k) + b_o u(k) + x_2(k),\\
x_2(k+1) &= -a_2y(k) + b_1u(k) + x_3(k),\\
&\vdots\\
x_n(k+1) &= -a_ny(k) + b_{n-1}u(k) + x_{n+1}(k),\\
x_{n+1}(k+1) &= b_nu(k).
\end{aligned} \tag{6.3.2}$$

Hence the state equation becomes:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \\ x_{n+1}(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ -a_n & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \\ x_{n+1}(k) \end{bmatrix} + \begin{bmatrix} b_o \\ b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u(k), \tag{6.3.3}$$

with the output equation expressed as

$$y(k) = [1\ 0\ \dots\ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n+1}(k) \end{bmatrix}. \tag{6.3.4}$$

The generic expressions for (6.3.3) and (6.3.4) are given by (Fig.6.3.2)

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \tag{6.3.5}$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k) + Du(k), \tag{6.3.6}$$

where

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ -a_n & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_o \\ b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix},$$

$$\mathbf{c}^T = [1\ \ 0\ \ \dots\ \ 0\ ], \qquad D = 0. \tag{6.3.7}$$

Equations (6.3.5 - 6.3.6) are called canonical form of state equations which is of special significance in system theory. Equations (6.3.3 - 6.3.4) are in observable canonical form.

The effect of variation of the time delay between y and u may be considered as follows.

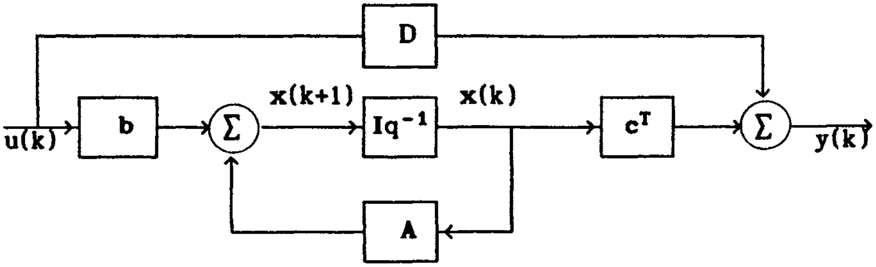


Figure 6.3.2 State-space diagram for the n-th order difference equation model (6.3.1).

Case 1 *Time delay = zero*

In (6.3.1), unity time-delay is assumed between the input u(k) and the output y(k). If the time delay is zero, that is

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n)$$
$$= b_ou(k) + b_1u(k-1) + \dots + b_nu(k-n),$$

the state-space representation (6.3.7) will be modified as

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} + \begin{bmatrix} b_1 - a_1b_o \\ b_2 - a_2b_o \\ \vdots \\ b_{n-1} - a_{n-1}b_o \\ b_n - a_nb_o \end{bmatrix} u(k), \qquad (6.3.8a)$$

with the output equation expressed as

$$y(k) = [1 \ 0 \ \dots \ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n+1}(k) \end{bmatrix} + b_ou(k). \qquad (6.3.8b)$$

Case 2 *Time-delay* = d

Now the process model (6.3.1) becomes

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n)$$
$$= b_ou(k-d) + b_1u(k-d-1) + \dots + b_nu(k-d-n),$$

and the state-space description changes to

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n+d}(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ -a_n & & & \ddots & \\ 0 & & & & 1 \\ \vdots & & 0 & & \\ 0 & & & & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n+d}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_o \\ \vdots \\ b_n \end{bmatrix} u(k),$$

and

$$y(k) = [1 \ 0 \ \dots \ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n+d}(k) \end{bmatrix}; \qquad (6.3.9)$$

here d-1 leading elements of **b** are 0, and D = 0, with reference to the model (6.3.5 - 6.3.6).

Remark: If the orders of difference polynomials in y and u in (6.3.1) are different, the dimension of the state descriptors are altered accordingly.

Advantages

Some general advantages of state-space modelling follow.

(1) *Internal modelling*
Since the internal physical dynamics can be modelled, better analysis and understanding of the overall input-output dynamics of the system are possible. Although some state variables may not have physical connotations, they may have analytical significance.

The model shows which states can be directly measured (or observed), and which can be controlled; not all states are observable or controllable. The model may be used to estimate the states which cannot be observed.

(2) *Simplification of expression*
A state-space model can be a simplified expression for a relatively complicated process, e.g., an n-th order difference equation model (6.3.1) can be represented by n first order difference equations in the state-space format (6.3.2 - 6.3.4). Similarly an n-th order differential equation can also be expressed as n first-order differential equations through state-space description.

(3) *Flexibility of modelling*
Since state variables are not unique, for the same input–output model, different state–space representations are possible.

(4) *Multivariable case*
The structural format of the state–space model is marginally altered when a multi–input multi–output process is considered instead of a single–input single–output process, which is computationally advantageous.

6.3.2 Processes with Noise

Equation (6.3.1) was a deterministic model in discrete time, where all variables were assumed to be known. In practice, the process may be subject to unknown disturbances acting as random input to the process or as noise, corrupting the output measurements; the former is referred to as the process noise and the latter as the measurement noise. The properties of the noise are discussed in Sec.2.3.1.

Consider the process model

$$
\begin{aligned}
y(k) + a_1 y(k-1) + \dots + a_n y(k-n) \\
&= b_o u(k-1) + b_1 u(k-2) + \dots + b_n u(k-n-1) \\
&\quad + e(k) + c_1 e(k-1) + \dots + c_n e(k-n), \qquad (6.3.10a)
\end{aligned}
$$

where u and y are the input and the output respectively as in (6.3.1) with e as the additional noise input to the system. The model (6.3.10a) can be concisely expressed as

$$
A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k), \qquad (6.3.10b)
$$

where d = 1 and

$$
A(q^{-1}) = 1 + a_1(q^{-1}) + a_2(q^{-1}) + \dots + a_n(q^{-1}),
$$

$$
B(q^{-1}) = b_o + b_1(q^{-1}) + b_2(q^{-1}) + \dots + b_n(q^{-1}),
$$

$$
C(q^{-1}) = 1 + c_1(q^{-1}) + c_2(q^{-1}) + \dots + c_n(\bar{q}^{1}\ ),
$$

The state–space model is given by

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n+1}(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ -a_n & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \\ x_{n+1}(k) \end{bmatrix} + \begin{bmatrix} b_o \\ b_1 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u(k) + \begin{bmatrix} c_1-a_1 \\ c_2-a_2 \\ \vdots \\ c_n-a_n \\ 0 \end{bmatrix} e(k), \tag{6.3.11a}$$

$$y(k) = [1\ 0\ \dots\ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_{n+1}(k) \end{bmatrix} + e(k). \tag{6.3.11b}$$

If the output measurements (or observations) are contaminated with additive noise (v), the measured output y^* can be expressed as

$$y^*(k) = y\ (k) + v(k). \tag{6.3.12}$$

The overall state-space and the observation model can be expressed as (Fig.6.3.3):

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k) + \mathbf{s}e(k), \tag{6.3.13}$$

$$y^*(k) \quad = \mathbf{c}^T\mathbf{x}(k) + e(k) + v(k); \tag{6.3.14}$$

A, **b**, **s** and **c** correspond to (6.3.11).

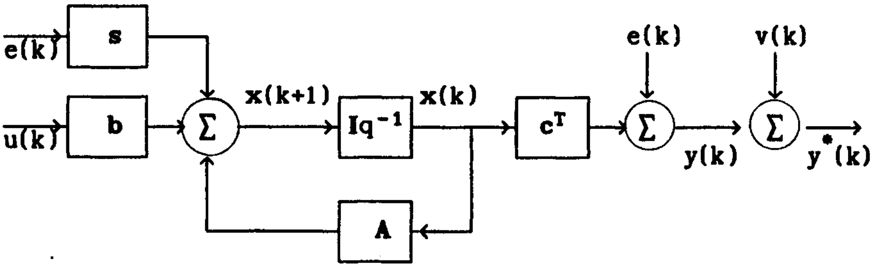


Figure 6.3.3 State-space diagram of an n-th order difference equation model (6.3.10) with noise input, and measurement noise at the output.

Remarks

(1) Note that any variable affecting the output directly (i.e. without any time delay) has to appear in the measurement equation (e.g., (6.3.14)) of the state-space model.

(2) If the process is completely deterministic, the noise terms in the process model and the state-space model will disappear. If the noise present is unmeasurable, its estimated values may be used:

$$\hat{e}(k) = y(k) - \mathbf{h}^T\hat{\mathbf{x}}(k|k-1),$$

where y(k) is the measured output and $\hat{\mathbf{x}}(k|k-1)$ is the one-step ahead prediction of the state, computed at time (k-1) (further discussions follow in Sec.6.6).

(3) The number of states required for minimal state-space representation (6.3.11) for an ARMAX process is given by max{δA, δB+d, δC), where δA, δB and δC are the degrees of $A(q^{-1})$, $B(q^{-1})$ and $C(q^{-1})$ respectively, and d is the time delay between exogenous input u and the output y.

Example 6.3.2(1) State-space model for a stage in the paper-making process

The basis weight (y) in the paper-making process, can be controlled by manipulating the thick stock flow (u); a typical model for the closed-loop controlled process is given by (Aström and Wittenmark, 1973)

$$\begin{aligned}(1-q^{-1})(1 - 1.283q^{-1} + 0.425q^{-2})y(k)\\ = (1-q^{-1})(2.307q^{-1} - 2.025q^{-2})u(k-2)\\ + 0.382(1 - 1.438q^{-1} + 0.550q^{-2})e(k);\end{aligned} \qquad (6.3.15)$$

The state-space model for this process can be produced as follows.

Rewriting (6.3.15),

$$\begin{aligned}(1 - 2.283q^{-1} + 1.708q^{-2} - 0.425q^{-3})y(k)\\ = (2.307q^{-1} - 4.332q^{-2} + 2.025q^{-3})u(k-2)\\ + (1 - 1.438q^{-1} + 0.550q^{-2})e'(k),\end{aligned}$$

where e'(k) = 0.382e(k). So the process (6.3.15) may be expressed as

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \end{bmatrix} = \begin{bmatrix} 2.283 & 1 & 0 & 0 & 0 \\ -1.708 & 0 & 1 & 0 & 0 \\ 0.425 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2.307 \\ -4.332 \\ 2.025 \end{bmatrix} u(k) + \begin{bmatrix} 0.845 \\ -1.158 \\ 0.425 \\ 0 \\ 0 \end{bmatrix} e'(k),$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \end{bmatrix} + e'(k).$$

Example 6.3.2(2) State-space model for the yearly averaged sunspot series

The subset-AR model for this series (Appendix 8A) is given in Example 3.6.3(1) as

$$y(k) = 1.2495y(k-1) - 0.551y(k-2) + 0.15y(k-9) + e(k).$$

The state-space model for (6.3.15) can be expressed as

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \\ x_6(k+1) \\ x_7(k+1) \\ x_8(k+1) \\ x_9(k+1) \end{bmatrix} = \begin{bmatrix} 1.2495 & 1 & 0 & \cdots & 0 \\ -0.5510 & 0 & 1 & \cdots & 0 \\ 0 & & & \ddots & 0 \\ 0 & & & & \\ 0 & & & & \\ 0 & & & \ddots & \\ 0 & 0 & & & \\ 0 & & & & 1 \\ 0.1500 & & & & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \\ x_7(k) \\ x_8(k) \\ x_9(k) \end{bmatrix} + \begin{bmatrix} 1.2495 \\ -0.5510 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.1500 \end{bmatrix} e(k),$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_9(k) \end{bmatrix} + e(k).$$

Remark: The state vector in case of the full AR model is of the same size as the one for the subset AR model.

6.4 STATE-SPACE MODELS FOR PERIODIC PROCESSES

A process or a time-series may have characteristic components like the trend, the periodic or seasonal component etc., which may be implicitly or explicitly modelled.

In the implicit approach, the trend and the seasonal component are absorbed along with the signal and the noise components into a general purpose transfer-function model through the appropriate differencing of the data as in the case of the Box and Jenkins models (Sec.4.3). This model can be expressed in state-space format as discussed in Sec.6.3.

In the explicit approach, a structural model (discussed in Sec.2.6.1) of the time series is presumed to be

$$y(k) = y_{tr}(k) + y_p(k) + \eta(k), \qquad (6.4.1)$$

where y_{tr} and y_p are the trend and periodic components respectively and η is the model uncertainty. It is assumed that the trend and the periodic component(s) are separately available; the composite series is modelled by modelling its trend and periodic components individually.

In this section explicit modelling has been considered.

6.4.1 The Trend Model

Consider a linear model for the trend:

$$\Delta y_{tr}(k+1) = \beta(k) + w_1(k+1), \qquad \Delta = 1-q^{-1},$$

that is

$$y_{tr}(k+1) = y_{tr}(k) + \beta(k) + w_1(k+1), \qquad (6.4.2)$$

where β represents the instantaneous change of level (i.e. the slope); it is assumed that β can be modelled as

$$\beta(k+1) = \beta(k) + w_2(k), \qquad (6.4.3)$$

where $\{w_1(k)\}$ and $\{w_2(k)\}$ are zero-mean, white noise sequences with known statistics:

$$E[\mathbf{w}_{tr}(i)\mathbf{w}_{tr}^T(j)] = \begin{cases} Q(i), & i = j, \\ 0, & i \neq j, \end{cases}$$

$$\mathbf{w}_{tr}(k) = [w_1(k) \;\; w_2(k)]^T.$$

The state-space representation for the trend component follows as

$$\mathbf{x}_{tr}(k+1) = \mathbf{A}_{tr}\mathbf{x}_{tr}(k) + \mathbf{S}_{tr}\mathbf{w}_{tr}(k), \qquad (6.4.4)$$

$$y_{tr}(k) \;= \mathbf{c}_{tr}^{T}\mathbf{x}_{tr}(k) + w_1(k), \qquad (6.4.5)$$

where

$$\mathbf{x}_{tr}(k+1) = \begin{bmatrix} x_{tr_1}(k+1) \\ x_{tr_2}(k+1) \end{bmatrix},$$

$$\mathbf{A}_{tr} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}, \qquad a_{11} = a_{12} = a_{22} = 1,$$

$$\mathbf{S}_{tr} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{c}_{tr} = [1 \;\; 0]^T.$$

So the states can be defined as

$$x_{tr_1}(k) = y_{tr}(k) - w_1(k),$$

$$x_{tr_2}(k) = \beta(k).$$

Example

The German unemployment series (Appendix 7E) has a trend component and a periodic component (see Fig. 14.3.1). Assuming the trend component to be separately available (say, through bidirectional filtering, discussed in Sec.14.3.1), the state-space model (6.4.4 - 6.4.5) may be used to represent the trend component.

The model is initialized with

$$x_{tr_1}(0) = y_{tr}(0), \quad \text{and}$$

$$x_{tr_2}(0) = \beta(0) = y_{tr}(1) - y_{tr_1}(0).$$

As the time index k increments, a new value for $y_{tr}(k)$ is available; based on the model (6.4.4 - 6.4.5), the estimate $\hat{y}_{tr}(k|k)$ may be produced using the Kalman filter discussed in Sec.6.6.2.

6.4.2 Periodic Component Model

There are different ways of modelling the periodic component.

(a) Basic model

Consider the model

$$\sum_{i=0}^{N-1} y_p(k-i) = w_p(k), \tag{6.4.6}$$

where N specifies the length of the period, and $\{w_p(k)\}$ is the zero-mean white noise sequence. Here it is assumed that the expectation of the periodic sums of the components y_p to be zero. The state-space representation is given by

$$\begin{aligned} x_p(k+1) &= A_p x_p(k) + g_p w_p(k), \\ y_p(k) &= c_p^T x_p(k) + w_p(k), \end{aligned}$$

where

$$x_p = \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \\ \vdots \\ x_{pT} \end{bmatrix}, \qquad A_p = \begin{bmatrix} -1 & -1 & \dots & -1 & -1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix},$$

$$g_p = \begin{bmatrix} -1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad c_p = [1\ 0\ \dots\ 0]^T.$$

(b) Periodic random walk model

The random walk process has been discussed in Sec.2.3.2. Periodic differencing may be used to model the periodic component as

$$y_p(k) - y_p(k-N) = w_p(k), \tag{6.4.7}$$

where N is the period length, and w_p is the zero mean white noise. The state-space representation follows as

$$
\begin{aligned}
\mathbf{x}_p(k+1) &= \mathbf{A}_p\mathbf{x}_p(k) + \mathbf{s}_p w_p(k),\\
y_p(k) &= \mathbf{c}_p^T\mathbf{x}_p(k) + w_p(k),
\end{aligned}
$$

where

$$
\mathbf{x}_p = \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \\ \vdots \\ x_{pT} \end{bmatrix}, \quad
\mathbf{A}_p = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ & 1 & \cdots & 0 & 0 \\ & & \ddots & \vdots & \vdots \\ & 0 & & 1 & 0 \end{bmatrix},
$$

$$
\mathbf{s}_p = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad \mathbf{c}_p = [1\ 0\ \ldots\ 0]^T.
$$

Example

Consider formation of the state-space model for (6.4.7) with N = 3.

Here,

$$
\begin{aligned}
\mathbf{x}_p(k+1) &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\mathbf{x}_p(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} w_p(k),\\
y_p(k) &= x_{p1}(k) + w_p(k).
\end{aligned}
$$

The validity of the model can be shown as follows.

$$
\begin{aligned}
x_{p1}(k+1) &= x_{p3}(k),\\
x_{p2}(k+1) &= x_{p1}(k) + w_p(k),\\
x_{p3}(k+1) &= x_{p2}(k).
\end{aligned}
$$

So

$$
\begin{aligned}
x_{p1}(k+1) &= x_{p2}(k-1),\\
&= x_{p1}(k-2) + w_p(k-2);
\end{aligned}
$$

hence

$$
y_p(k+1) = y_p(k-2) + w_p(k+1),
$$

which is the same as the model (6.4.7).

(c) Periodic model based on trigonometric functions

The trigonometric Fourier series representation of a periodic signal is discussed in Sec.2.5.1. and Sec.2.5.3.

If the periodic sequence $\{y_p(k)\}$ has a period-length N,

$$y_p(k) = a_o + \sum_{n=1}^{M} [a_n\cos(n\omega k) + b_n\sin(n\omega k)], \qquad (6.4.8)$$

where if N is odd, M = (N-1)/2, and if N is even M = N/2; for n = N/2,

$$\cos(n\omega k) = (-1)^k, \text{ and } \sin(n\omega k) = 0.$$

Let N be even (see Remarks for the case when N is odd). The coefficients a_n and b_n are given by

$$a_o = \frac{1}{N}\sum_{k=1}^{N} y_p(k),$$

$$a_n = \frac{2}{N}\sum_{k=1}^{N} y_p(k)\cos(n\omega k),$$

$$b_n = \frac{2}{N}\sum_{k=1}^{N} y_p(k)\sin(n\omega k), \qquad (6.4.9)$$

where $n = 1, 2,\dots, \frac{N}{2}-1$. For $n = \frac{N}{2}$,

$$a_{N/2} = \frac{1}{N}\sum_{k=1}^{N} y_p(k)(-1)^k. \qquad (6.4.10)$$

The objective is to estimate a_n and b_n defined as the states, while the orthogonal basis functions given by the sine and cosine terms are known. The recursive formulation for the coefficients, a_n and b_n, is developed as follows.

Define

$$y(k) = a_o(i) + \sum_{n=1}^{\frac{N}{2}-1} [a_n(i)\cos n\omega(k-i) + b_n(i)\sin n\omega(k-i)] + a_{N/2}(i)(-1)^{(k-i)}, \qquad (6.4.11)$$

i being an arbitrary time index; equation (6.4.11) is valid because the set of time functions,

$$\{1 \ \cos\omega(k-i) \ \sin\omega(k-i) \ \cos2\omega(k-i) \ \sin2\omega(k-i) \ (-1)^{(k-i)}\},$$

is a set of orthogonal basis functions for any series which is periodic with period N. Following (6.4.11), y(k) can be redefined as

$$y(k) = a_o(i+1) + \sum_{n=1}^{\frac{N}{2}-1} [a_n(i+1)\cos n\omega(k-i-1) + b_n(i+1)\sin n\omega(k-i-1)] + a_{N/2}(i+1)(-1)^{(k-1-1)}. \tag{6.4.12}$$

Now,

$$\cos n\omega[(k-i)-1] = \cos n\omega(k-i)\cos n\omega + \sin n\omega(k-i)\sin n\omega,$$

$$\sin n\omega[(k-i)-1] = \sin n\omega(k-i)\cos n\omega - \cos n\omega(k-i)\sin n\omega.$$

Hence equating the coefficients of $\cos n\omega(k-i)$ and $\sin n\omega(k-i)$ in (6.4.11) and (6.4.12),

$$a_n(i+1)\cos n\omega - b_n(i+1)\sin n\omega = a_n(i),$$

$$b_n(i+1)\cos n\omega + a_n(i+1)\sin n\omega = b_n(i).$$

Since

$$\begin{bmatrix} \cos n\omega & -\sin n\omega \\ \sin n\omega & \cos n\omega \end{bmatrix} = U_n^T \text{ (say)}, \tag{6.4.13}$$

is an orthogonal matrix (i.e. $U_n^T U_n = U_n U_n^T = I$),

$$\begin{bmatrix} a_n(i+1) \\ b_n(i+1) \end{bmatrix} = \begin{bmatrix} \cos n\omega & \sin n\omega \\ -\sin n\omega & \cos n\omega \end{bmatrix} \begin{bmatrix} a_n(i) \\ b_n(i) \end{bmatrix}. \tag{6.4.14}$$

Again, from (6.4.11) and (6.4.12),

$$a_o(i+1) = a_o(i),$$

$$a_{N/2}(i+1) = (-1)a_{N/2}(i). \tag{6.4.15}$$

Hence with the states defined as

$$x(i) = [a_o(i) \quad a_1(i) \quad b_1(i) \quad a_2(i) \quad b_2(i) \ldots \quad a_{N/2}(i)]^T, \tag{6.4.16}$$

$$x(i+1) = A_p x(i), \tag{6.4.17}$$

where

$$A_p = \begin{bmatrix} 1 & & & & \\ & U_1 & & & \\ & & U_2 & & \\ & & & \ddots & \\ & & & & -1 \end{bmatrix}. \tag{6.4.18}$$

Again in (6.4.11), for $i = k$, the cosine terms become unity and the sine terms become zero, that is

$$y(k) = a_o(k) + \sum_{n=1}^{\frac{N}{2}-1} a_n(k) + a_{N/2}(k). \qquad (6.4.19)$$

The state-space representation for the seasonal component is therefore given by

$$x(k+1) = A_p x(k) + w_p(k), \qquad (6.4.20)$$

$$y(k) \quad = c_p^T x(k) + v_p(k), \qquad (6.4.21)$$

where following (6.4.16) and (6.4.19),

$$c_p = [1\ 1\ 0\ 1\ 0\ \ldots\ 1\ 0\ 1]^T,$$

v_p and w_p are measurement noise and the seasonal process noise respectively; A_p is given by (6.4.18).

Remarks

(a) A_p in (6.4.18) is an orthogonal matrix.

(b) The first term $a_o(k)$ in (6.4.11) represents the average level; it can be eliminated if $\{y(k)\}$ is zero-mean.

(c) The period length N was assumed to be even here. If N is odd, the term $a_{N/2}$ as in (6.4.10) will disappear; the rest of the formulation is similar.

Example 6.4.2 Modelling the periodic component in the German unemployment series

This is a series of monthly data with yearly periodicity (Appendix 7E), N = 12. It is assumed that trend and the periodic components of the series are separately available (see Sec.14.3.1). The modelling of the trend component is discussed in Sec.6.4.1; the periodic component is assumed to be available through bidirectional filtering as in Example 14.3.1. The modelling starts from the year 1950.

(i) Define the state vector (6.4.16) of length N

$$x(k) = [a_o\ a_1\ b_1\ a_2\ b_2\ a_3\ b_3\ a_4\ b_4\ a_5\ b_5\ a_6]^T.$$

(ii) The state-space model is given by (6.4.20 - 6.4.21).

(iii)i A_p is given by

$$A_p = \begin{bmatrix} 1 & & & & & & \\ & U_1 & & & & & \\ & & U_2 & & & & \\ & & & U_3 & & & \\ & & & & U_4 & & \\ & & & & & U_5 & \\ & & & & & & -1 \end{bmatrix},$$

where

$$U_i = \begin{bmatrix} \cos i\omega & \sin i\omega \\ -\sin i\omega & \cos i\omega \end{bmatrix}, \quad i = 1 \text{ to } 5,\ \omega = 2\pi/12.$$

(iv) c_p is given by

$$c_p = [1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]^T$$

(v) x(0) is given by (6.4.9) (n = 1 to 5, N = 12, $\omega = 2\pi$); the initial state is assumed to start from 1950. For the first period (k = 1) the values of initial states ($\times 10^{-5}$) work out to be

a_0=2.582, a_1=0.549, a_2=−0.021, a_3=0.007, a_4=−0.112,

a_5=0.046, a_6=0.005,

b_1=3.842, b_2=1.334, b_3=0.523, b_4=0.183, b_5=−0.055

(vi) Assume **Q** and R are the covariances of the process noise w_p and the measurement noise v_p respectively. The initial state-estimation error covariance $\mathbf{P}_o$ is also assumed. A typical choice could be $\mathbf{Q} = 10\mathbf{I}$, $R = 1$, $\mathbf{P}_o = 100\mathbf{I}$.

Having developed the state-space model, the Kalman filter (Sec.6.6.2) can be used to obtain the optimal estimates of the available observations {y(k)}.

Remarks

(a) The process and measurement noise covariances **Q** and R can be estimated (Mehra, 1972), but the procedure is quite complex. So a trial and error approach may be applied, which however is a drawback.

(b) A choice of $\mathbf{P}_o$ with high diagonal values implies low confidence in the initial estimates of the states.

(c) Steps (i) to (iv) are common to all time-series with monthly data and yearly periodicity.

6.4.3 Prediction Problem Formulation

For a series with additive structural components (6.4.1), the combined state-space representation is given by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{w}(k),$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k) + v(k),$$

where

$$x(k) = \begin{bmatrix} x_{tr}(k) \\ x_p(k) \end{bmatrix}, \qquad A = \begin{bmatrix} A_{tr} & 0 \\ 0 & A_p \end{bmatrix}, \qquad c = [c_{tr}\ c_p]^T,$$

$$w(k) = [w_{tr}(k)\ w_p(k)]^T, \qquad v(k) = v_{tr}(k) + v_p(k).$$

All the components are specified by the trend model and the periodic process model discussed in Sec.6.4.1 and Sec.6.4.2 respectively. The subsequent estimation of the states and multistep prediction will involve the following steps.

(a) Assumption of the noise statistics: the mean and the covariance of w and v need to be assumed. Usually these noise processes are assumed to be zero-mean, Gaussian.

(b) Initialization of the state estimates: this can be obtained from (6.4.9) and (6.4.10).

(c) State estimation: use an optimal estimator to compute recursively the state estimates $\hat{x}(k|k)$. The Kalman filter, discussed in Sec.6.6, may be used.

(d) Multistep prediction: the p-step ahead prediction is given by

$$\hat{y}(k+p|k) = c^T\hat{x}(k+p|k),$$

where

$$\hat{x}(k+p|k) = A^p\hat{x}(k|k),$$

as discussed in Sec.6.7.

6.5 OPTIMAL STATE ESTIMATION

The states of a system can be estimated from available measurements which may be contaminated with noise.

One of the ways of optimal estimation of the states is the Bayesian approach. The Bayesian method considers the variables to be estimated as random, and it requires knowledge of the prior probability distribution function of these variables (regarded as parameters). The Bayesian rule states that the conditional probability density function $P(x|y)$ is given by

$$P(x|y) = \frac{P(x)P(y|x)}{\int P(x)P(y|x)dx},$$

where the random variable x is observed through the noisy measurement y; *P*(x) is the prior probability density function of x before y is observed, and the likelihood function *P*(y|x) specifies the probabilities of y observations as a function of a range of possible values of x (in the present context, x refers to the parameter values). The main problems in the realization of Bayesian estimator are:

(a) the prior statistics of the variables to be estimated may not be available, whereas in the case of incorrect assumptions of the prior statistics *P*(x), the estimated parameters are likely to be biased.

(b) this method requires evaluation of the posterior probability distribution function *P*(y|x), which can be computationally intensive.

A statistical alternative to the probabilistic approach is to consider the Gauss-Markov model:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k) + \mathbf{w}(k),$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k),$$

where the process noise **w** and the measurement noise **v** have Gaussian or normal distributions; in such cases the states can be estimated using the Kalman filter.

6.6 THE KALMAN FILTER

The Kalman filter (Kalman, 1960) is the best linear estimator which can produce an optimal estimate of the states of a linear dynamic system, subject to the disturbances having Gaussian distribution. The optimality is in minimum estimation-error-variance sense.

In general, the function of a *filter* is to separate the signal or information from the noise-corrupted data. In the field of signal processing, filtering usually refers to frequency domain separation of signal from noise, e.g., low-pass, high-pass or band-pass filtering. The separation is possible only if the signal and the noise lie in different frequency ranges. When the signal and the noise occupy overlapping frequency bands, the Kalman filter can be used to compute the optimal estimate of the signal.

6.6.1 The Estimation Problem

Consider the following difference equation representation of the system and its environment in a multi-input multi-output configuration

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{S}\mathbf{w}(k), \tag{6.6.1}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k), \tag{6.6.2}$$

where **x** is nx1 state vector, **y** is the rx1 vector of measured outputs, **w** is the px1 vector of random inputs to the system ($p \le n$) and **v** is the rx1 vector of additive measurement noise. The matrices **A**, **S** and **C** are nxn, nxp and rxn matrices respectively. Here, the deterministic input u is assumed to be zero, which is not a limitation.

The objective is to obtain $\hat{\mathbf{x}}(k|k)$, the minimum variance estimate based on the available information on the states $\hat{\mathbf{x}}(k|k-1)$ and the measurements, $\mathbf{y}(0),\ldots,\mathbf{y}(k)$, subject to the following assumptions.

(a) Noise statistics

The process noise $\{\mathbf{w}(k)\}$ and the measurement noise $\{\mathbf{v}(k)\}$ are independent, zero-mean, white noise sequences with known statistics :

$$E[\mathbf{w}(k)] = 0\ , \quad \text{for all } k, \tag{6.6.3}$$

$$E[\mathbf{v}(k)] = 0\ , \quad \text{for all } k, \tag{6.6.4}$$

$$E[\mathbf{w}(k)\mathbf{w}^T(\tau)] = \begin{cases} \mathbf{Q}(k) & k = \tau, \\ 0 & k \neq \tau, \end{cases} \tag{6.6.5}$$

$$E[\mathbf{v}(k)\mathbf{v}^T(\tau)] = \begin{cases} \mathbf{R}(k) & k = \tau, \\ 0 & k \neq \tau, \end{cases} \quad \text{and} \tag{6.6.6}$$

$$E[\mathbf{w}(k)\mathbf{v}^T(\tau)\ ] = 0 \qquad \text{for all } k,\ \tau. \tag{6.6.7}$$

Remark: The Kalman filter equations (Sec.6.6.2) can be proved to be valid even if the noise sequences are not uncorrelated and (6.6.7) is violated.

(b) Initial States

The initial state $\mathbf{x}(0)$ is a random variable with known statistics:

$$\text{mean:} \quad E[\mathbf{x}(0)] = \mathbf{x}_o, \quad \text{and}$$

$$\text{variance:} \quad E[(\mathbf{x}(0) - \mathbf{x}_o)(\mathbf{x}(0) - \mathbf{x}_o)^T] = \mathbf{P}_o. \tag{6.6.8a}$$

The initial state $\mathbf{x}(0)$ is assumed to be independent of the process noise $\{\mathbf{w}(k)\}$:

$$E[\mathbf{x}(0)\mathbf{w}^T(k)] = 0 \quad \text{for all } k \geq 0. \tag{6.6.8b}$$

(c) System parameters
The parameters of **A**, **S** and **C** are assumed to be known.

The Kalman filter produces the best estimate $\hat{\mathbf{x}}(k|k)$ of $\mathbf{x}(k)$, minimizing the scalar cost function

$$J = E[(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))^T(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))].$$

6.6.2 Kalman Filter Equations

The state estimate and one-step prediction are given by the following equations.

(i) *Measurement update (or correction)* equations:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1)]. \tag{6.6.9}$$

(ii) *Time update (or prediction)* equations:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k). \tag{6.6.10}$$

The process and the measurement models (6.6.1–6.6.2) and the Kalman state estimator can be schematically expressed as shown in Fig.6.6.1.

The weighting or gain matrix **K** in (6.6.9) is given by

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{C}^T[\mathbf{C}\mathbf{P}(k|k-1)\mathbf{C}^T + \mathbf{R}(k)]^{-1}, \tag{6.6.11}$$

where **P**, the covariance of the estimation error for the measurement update and the time update, is given by

$$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{C}]\ \mathbf{P}(k|k-1), \tag{6.6.12}$$

$$\mathbf{P}(k+1|k) = \mathbf{A}\mathbf{P}(k|k)\mathbf{A}^T + \mathbf{S}\mathbf{Q}(k)\mathbf{S}^T, \tag{6.6.13}$$

the initial conditions at $k = k_o$ being

$$\hat{\mathbf{x}}(k_o|k_o-1) = \mathbf{x}_o, \text{ and } \mathbf{P}(k_o|k_o-1) = \mathbf{P}_o.$$

Thus, following (6.6.9) to (6.6.13), with the progression of

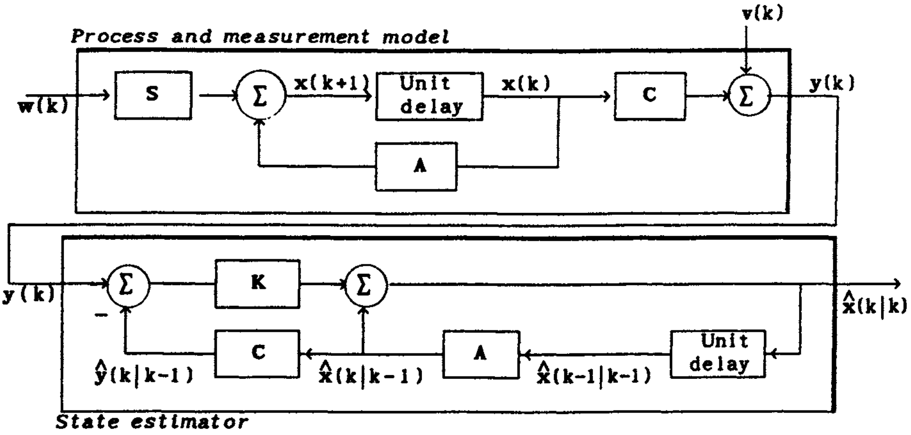
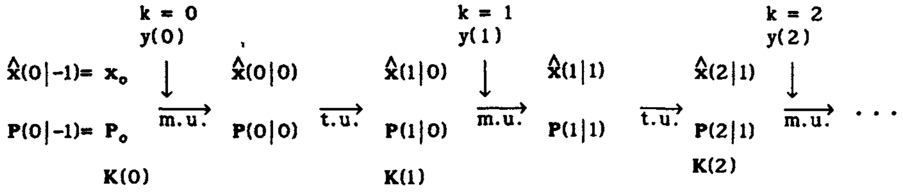


Figure 6.6.1 Structural diagram of the process and the measurement model, coupled with the Kalman filter state estimator.



m.u. measurement update.

t.u. time update.

Figure 6.6.2 Flow diagram of the sequential computation of the state estimates, the covariance matrix and the Kalman (filter) gain.

time, as new measurements y(k) become available, the estimation and the prediction of the states, and updating the corresponding error covariances and the Kalman gain can follow recursively through the two stages of measurement update and time update as shown in Fig.6.6.2.

6.6.3 Properties and Salient Features

(a) *Innovations sequence*

The term $C\hat{x}(k|k-1)$ in (6.6.9) is the one-stage predicted output $\hat{y}(k|k-1)$, and $\{y(k) - C\hat{x}(k|k-1)\}$ is the one-stage prediction error sequence, also referred to as the *innovations sequence*. $K(k)[y(k) - C\hat{x}(k|k-1)]$, the weighted innovation, acts as a correction to the predicted estimate $\hat{x}(k|k-1)$ to form the estimation $\hat{x}(k|k)$; the weighting matrix K is commonly referred to as the filter gain or the *Kalman gain* matrix.

The innovation represents the additional information available to the filter in consequence to the new observation y(k). For an optimal filter, the innovation sequence is a sequence of independent Gaussian random variable.

The occurance of bad data first shows up in the innovation vector. The discrepancies in the innovations sequence may be analysed probabilistically and if bad data is identified, it can be discarded before it can damage the filter

(b) *Estimation error covariances*

In the course of Kalman filtering, the estimation error covariances (6.6.12 - 6.6.13) indicate the degree of inaccuracy of the estimates. The error covariances are dependent on the process parameters and the noise statistics but not on the measurements. So different state-space designs may be Kalman filtered to ascertain which model produces least estimation error covariances and hence produces best estimates. Estimation error covariances may also be used to determine if the uncertainty in the estimates can be improved by the introduction of additional states or additional sensors.

(c) *Sampling periodicity and data reconstruction*

The sampling interval need not be uniform for the operation of the Kalman filter. There are two possibilities:

(i) Normal operation: even if the time intervals between subsequent measurements vary, the Kalman filter may be run as normal, i.e. iterating through the twin steps of measurement update and time update; the next measurement update is performed only when a new measurement is received, and the time update follows, and so on.

(ii) Data reconstruction: until the next measurement is available the state estimation may keep cycling continuously only through the time update (6.6.10). Thus

$$\hat{x}(k_2|k_1) = A(k_2|k_1)\hat{x}(k_1|k_1),$$

where

$$A(k_2|k_1) = A^{k_2-k_1};$$

for further details see (6.7.5). k_1 is the time when the last measurement was available. This is an interesting data reconstruction feature of the Kalman filter, which can be utilized for multistep prediction (discussed in Sec.6.7), as well as for reconstruction of the missing data.

(d) *Other aspects*

(i) Since the Kalman filter is a linear filter, moments up to the second order (i.e. means and covariances) appear in the filter formulations (6.6.9 - 6.6.13). (ii) The application of the Kalman filter for data or state smoothing is discussed in Sec.14.2.

Adaptive Kalman filtering

In general, the adaptation may be with respect to unknown, slowly time-varying noise covariances, deterministic inputs, parameter values etc. Since Kalman filtering mainly concerns estimation of states in a noisy environment, adaptive Kalman filtering usually refers to adaptation with respect to noise statistics.

When the state-space model is known, but the process noise and the measurement noise covariances **Q** and **R** in (6.6.5 - 6.6.6) are unknown, adaptive filtering can be performed using the Bayesian, the maximum likelihood, the correlation or the covariance matching methods (Mehra, 1972). The noise is assumed to be stationary. Adaptation with respect to deterministic but unknown inputs, or with respect to both unknown noise covariances and unknown deterministic inputs, is also possible; see for example Moghaddamjoo (1989). The problem of adaptation with respect to unknown parameters has bęen discussed in Chapter 3. Adaptive simultaneous estimation of parameters and the states can be performed using extended Kalman filters, where the parameter vector is appended to the state vector, and the enlarged state vector is estimated (Jazwinski, 1970, Anderson and Moore, 1979).

6.6.4 Implementation Aspects

Initialization

For the implementation of the Kalman filter, the following should be known:

(a) the parameters constituting the matrices **A**, **S** and **C** in the model (6.6.1 - 6.6.2).

(b) the covariances of the noise **Q** and **R** in (6.6.5 - 6.6.6).

(c) the initial estimates of the state $\mathbf{x}_0$, and the initial covariance of estimation error $\mathbf{P}_0$.

If any of the above are unknown, they have to be assumed or estimated. The parameters of the model can be estimated, for example using the least squares method from the transfer-function model (say, AR or ARMA model) of the process using the available input-output data. The noise covariances **Q** and **R** can be computed from the available data and prior true or approximate knowledge of the states. They may also be estimated (see remarks on adaptive Kalman filtering, above). The initial value of the state $\mathbf{x}_0$ may be directly known from the available measurements or may be assumed based on prior process knowledge. The estimation error covariance $\mathbf{P}_0$ is indicative of the confidence on the initial state estimate, and hence can be appropriately chosen. Usually $\mathbf{P}_0$ is considered to be a diagonal matrix.

The Kalman filter is not sensitive to incorrect choice of noise covariances **Q** and **R**. However it is necessary that the model is representative for the Kalman filter to produce sensible estimates.

Computation

Instead of updating the symmetric matrices **P** in (6.6.12 - 6.6.13), its square root factor **M** may be updated, where

$$\mathbf{P} = \mathbf{M}\mathbf{M}^T.$$

Some advantages of square-root filtering are as follows.

(a) Due to computational errors, on updating, **P** may not work out to be non-negative definite, leading to erroneous results, whereas $\mathbf{M}\mathbf{M}^T$ always has to be non-negative definite, thus offering better numerical stability.

(b) Numerical inaccuracy due to machine round off etc. pertaining to the square root factor **M** will be less than that of **P**.

There are different approaches to square root filtering (Bierman, 1977). One of the popular approaches is UD factorization:

$$P = UDU^T,$$

where **U** is upper triangular, and **D** is a diagonal matrix. In this approach there is no need for any square root operation. So instead of **P**, the **U** and **D** factors are sequentially updated. In the present case UD-measurement update (6.6.12) and UD-time update (6.6.13) are to be computed, as discussed in Appendix 3A and Appendix 13C respectively.

Summary of sequential implementation:

(1) Initialize the Kalman filter with the parameters of models (6.6.2 and 6.6.3), and **Q**, **R**, x_0, P_0.

(2) Read in a new measurement vector **y**.

(3) Perform measurement updates, (6.6.9) and (6.6.12), and compute $\hat{x}(k|k)$ and $P(k|k)$.

(4) Perform time updates (6.6.10) and (6.6.13), and compute $\hat{x}(k+1|k)$, $P(k+1|k)$ and $K(k+1)$.

(5) Go to (2).

6.7 OPTIMAL PREDICTION

This section concerns optimal multistep prediction of the state variable $\hat{x}(k+p|k)$; if desired, the prediction of the output variables can be obtained from the predicted states following the measurement equation (6.6.2) as

$$\hat{y}(k+p|k) = C\hat{x}(k+p|k).$$

The concept of prediction is inherent with the Kalman filter, and it has already been shown that one-step ahead prediction of the state, $\hat{x}(k+1|k)$ is generated by the Kalman filter algorithm (6.6.9 - 6.6.10). Multistep prediction follows as a natural extension.

The objective is to produce p-step ahead estimate $\hat{x}(k+p|k)$ of **x** at time k, given all data up to time k, where $p \geq 1$. The prediction will be optimal in a minimum variance sense, subject to the minimization of the cost

$$J = E\ \{(x(k+p) - \hat{x}(k+p|k))^T(x(k+p) - \hat{x}(k+p|k))\}.$$

Following (6.6.2 - 6.6.3), consider the dynamic process and measurement models

$$\hat{x}(k+1) = Ax(k) + Bu(k) + Sw(k), \tag{6.7.1}$$

$$y(k) = Cx(k) + v(k), \tag{6.7.2}$$

where **u** is a hx1 deterministic input vector and **B** is a nxh matrix which is assumed to be known; the other assumptions stated in Sec.6.6.1 relating to the model (6.6.2 - 6.6.3) are assumed to be valid.

Following (6.7.1), multistep prediction may be obtained as

$$\begin{aligned} x(k+2) &= Ax(k+1) + Bu(k+1) + Sw(k+1) \\ &= A^2x(k) + ABu(k) + Bu(k+1) + ASw(k) + Sw(k+1). \end{aligned}$$

Similarly,

$$\begin{aligned} x(k+3) &= Ax(k+2) + Bu(k+2) + Sw(k+2) \\ &= A^3x(k) + A^2Bu(k) + ABu(k+1) + Bu(k+2) \\ &\qquad + A^2Sw(k) + ASw(k+1) + Sw(k+2). \end{aligned}$$

Hence

$$x(k+p) = A^px(k) + \sum_{i=k}^{m} A^{m-i}Bu(i) + \sum_{i=k}^{m} A^{m-i}Sw(i), \tag{6.7.3}$$

where m = k+p-1. Hence the predicted states based on the available measurements upto time k is given by

$$\begin{aligned} \hat{x}(k+p|k) &= E\{x(k+p)|y(0), y(1),\dots,y(k)\} \\ &= A^pE\{x(k)|y(0), y(1),\dots,y(k)\} \\ &\qquad + \sum_{i=k}^{m} A^{m-i}B\, E\{u(i)|y(0), y(1),\dots,y(k)\} \\ &\qquad + \sum_{i=k}^{m} A^{m-i}S\, E\{w(i)|y(0), y(1),\dots, y(k)\}. \end{aligned} \tag{6.7.4}$$

Since the random vectors {w(k), w(k+1),..., w(k+p-1)} are independent of {y(0), y(1),..., y(k)}, the last term in (6.7.4) vanishes. The optimal p-step state predictor follows as:

$$\hat{x}(k+p|k) = A^p\hat{x}(k|k) + \sum_{i=k}^{m} A^{m-i}Bu(i). \tag{6.7.5a}$$

The optimal predictor can also be expressed as

$$\hat{x}(k+p|k) = A(k+p|k)\hat{x}(k|k) + \sum_{i=k}^{m} A(m|i)Bu(i), \tag{6.7.5b}$$

where $A(k+j|k) = A^j$.

The prediction error is given by,

$$\tilde{x}(k+p|k) = x(k+p) - \hat{x}(k+p|k)$$

$$= A^p[x(k) - \hat{x}(k|k)] + \sum_{i=k}^{m} A^{m-i}Sw(i),$$

that is

$$\tilde{x}(k+p|k) = A^p\tilde{x}(k|k) + \sum_{i=k}^{m} A^{m-i}Sw(i). \qquad (6.7.6)$$

It can be shown (see Appendix 6) that the covariance matrix corresponding to the optimal p-step prediction is given by,

$$P(k+p|k) = A^pP(k|k)(A^p)^T + \sum_{i=k}^{m} A^{m-i}BQ(i)B^T(A^{m-i})^T, \qquad (6.7.7)$$

where m = k+p–1.

6.8 CASE STUDY: ESTIMATION AND PREDICTION OF INGOT TEMPERATURES AND HEATING IN SOAKING PITS

A typical application of the Kalman filter in the steel-making industry is studied here.

Prior to rolling, steel ingots have to be uniformly heated to a specific temperature, which is often achieved through heating in furnaces, called *soaking pits*. The Kalman filter may be used to estimate the inaccessible ingot-surface and the ingot-centre temperatures, and thereby the time when the ingots will be available for rolling can be predicted; this information is of vital importance for efficient plant management as well as for saving fuel.

6.8.1 Process Description and Problem Statement

Molten steel is poured into cast iron moulds to form ingots. As the mould cools naturally in air, the ingot solidifies inwardly from the surfaces, shrinking up to approximately 7% by volume. When sufficiently cooled, the train of cars carrying the moulds is moved to the stripping yard, where the ingots are freed from the moulds; the centre of the ingot may still be molten. The train of the hot ingots is then moved to the soaking pits.

A soaking pit (Fig.6.8.1(a)) is a rectangular chamber furnace with an arrangement for loading ingots through the sliding top cover. Gaseous or liquid fuel is fired in

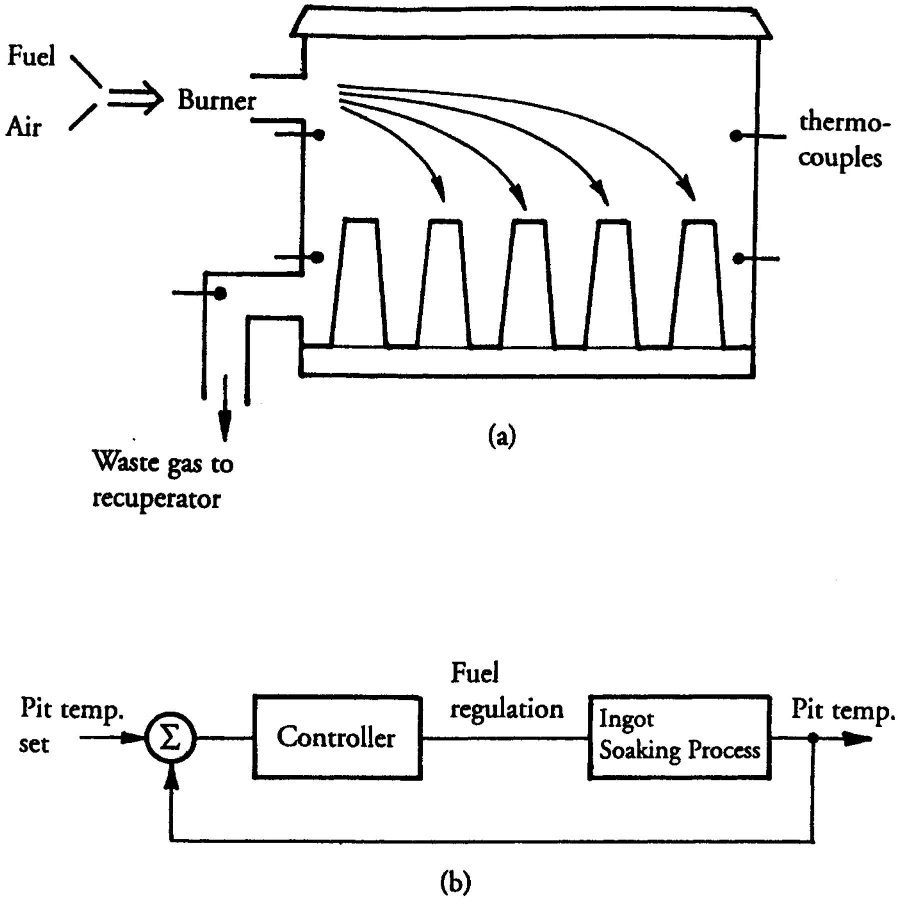


Figure 6.8.1 (a) Schematic diagram of the ingot heating process in a soaking pit.
(b) The pit temperature control system.

the furnace and the ingots are heated largely by radiation from the products of combustion and the refractory pit walls. The pit temperature control schematic is shown in Fig.6.8.1(b). Typically 5-20 hot or cold ingots, each 5 to 20 tons by weight, may be thermally soaked in the pit at one time. The temperature of the cold incoming ingots will be the same as the ambient temperature, whereas temperature distribution for hot ingots from the centre to the surface

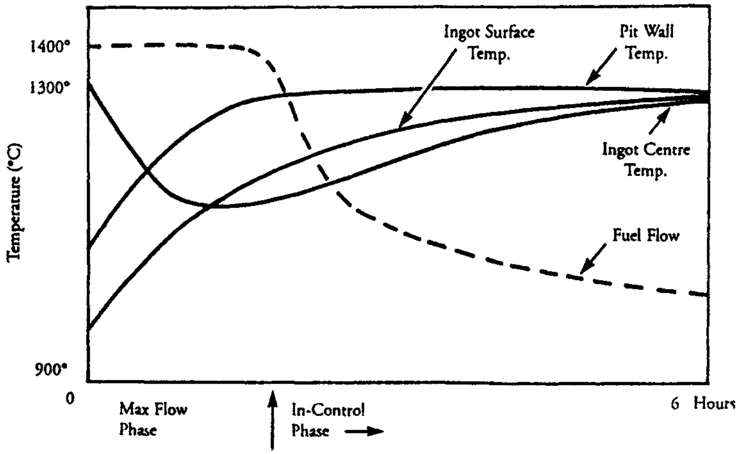


Figure 6.8.2 Typical patterns of the pit and the ingot temperatures and the fuel flow.

may range between, 1200-1600°C to 800-1000°C. The ingots are heated and soaked for several hours (say, 4-8 hours) to bring them to the desired temperature (typically, 1280°C), uniform throughout the cross-section. The ingots are said to be soaked through when

(a) complete solidification of the centre has taken place,

(b) the difference between the surface temperature and the core temperature reaches within a specified value, and

(c) the average temperature is greater than a given value.

The heating in the soaking pit is carried out in two phases (see Fig.6.8.2).

(i) *Maximum flow phase*: this is the the initial phase, when fuel is supplied at the maximum permissible rate to the furnace, until the furnace temperature reaches a preset value,

(ii) *In-control phase*: during this phase, the temperature controller regulates the fuel flow to maintain the furnace temperature (measured by the wall thermocouples) at a set value. The fuel flow rate gradually drops and the waste gas temperature rises until the ingots are completely soaked, when both the fuel flow rate and the waste gas temperature tend to stabilize.

While under-soaking of the ingots results in bad rolling, increased rejection and possible roll damage, over-soaking is wasteful from pit utilization and fuel optimiza-

optimization points of view.

The problem

The main objectives are

(i) estimation of the ingot-centre and surface temperatures, and

(ii) prediction of the *ready-to-roll time.*

The estimation and the prediction become difficult because of the large number of uncontrollable and unmeasurable factors influencing the processes, some of which are as follows.

(a) The ingot characteristics: for example, the time between the teeming, stripping and charging of the ingots, the atmospheric conditions, and the ingot size and composition may be widely varying.

(b) The pit characteristics: for example, the heating dynamics, the charge distribution, bottom buildup due to scale accumulation inside the pit, recuperator efficiency etc. may keep changing.

Hence prediction of ready-to-roll time based on a heat transfer model alone will be difficult. Kalman filtering approach can be useful under such circumstances.

The use of Kalman filter for estimation and prediction of the soaking pit process is discussed in many papers, for example Lumelsky (1983), Wick (1982). The present study is mainly based on the work of Lumelsky (1983). The ingot soaking process is expressed by a linear state-space model. The same model is used for the maximum-flow phase and the in-control phase of the soaking pit process. A heat-transfer model is used to generate data on the ingot-surface and the ingot-centre temperatures; these data together with the pit wall temperature measurement are used for the estimation of the parameters. The ingot-surface and the ingot-centre temperatures are estimated and predicted using the Kalman filter.

6.8.2 Modelling, Estimation and Prediction

Process model

The ingot-soaking process is represented by the model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k) + \mathbf{w}(k), \quad (6.8.1)$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k) + v(k), \quad (6.8.2)$$

where

$$\mathbf{x}(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix},$$

x_1, the ingot-surface temperature,
x_2, the ingot-centre temperature,
x_3, the pit wall temperature,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \quad \mathbf{c} = [0\ 0\ 1]^T;$$

so the measurement, $y = x_3$, the pit wall temperature, and the control variable u is the fuel flow. **w** and v are the process noise and the measurement noise respectively.

From operational considerations (explained below), $a_{23} = a_{32} = 0$ and $b_2 = 0$.

Remarks

(1) The assumption, $a_{23} = a_{32} = b_2 = 0$, implies that the ingot-centre temperature is a function of the ingot-surface temperature.

(2) The model (6.8.1 - 6.8.2) is a simplified model. Additional variables may be incorporated, if necessary. For example, the waste gas temperature can be used as an additional output measurement.

(3) The pit wall temperature is usually the average of the two measurements obtained from the thermocouples located on the burner wall and the wall opposite, to minimize the effect of nonuniform heating inside the pit.

Parameter estimation

To estimate the parameters of the model (6.8.1), it is necessary to know the ingot-surface and the ingot-centre temperatures, which can be obtained from heat transfer models of the pit and the ingot; such models are discussed in Massey and Sheridan (1971), Hinami *et al* (1975), Kung *et al* (1967). Given the initial and operating conditions, e.g., ingot history, heating strategy etc., the temperature profiles of the ingot-surface, ingot-centre and hence the eventual ready-to-roll time can be computed using the heat transfer models. Based on the ingot-surface and the ingot-centre temperatures obtained from the model, the parameters in (6.8.1) can be determined using the least squares method.

Initialization of the Kalman filter

Once the parameters are known, the Kalman filter can be used to obtain the state estimates. The Kalman filter is initialized with assumed values for the states x_1, x_2, x_3, the process noise and the measurement noise covariances, and the estimation error covariances. It is found that the eventual performance of the Kalman filter is not very sensitive to imprecise assumptions at initialization.

The Kalman filter (6.6.9 - 6.6.13) is used for the state estimation (i.e. estimation of the ingot-surface and the ingot-centre temperatures), and the optimal predictor (6.7.4) is used for ready-to-roll-time prediction. Some of the filter and the predictor equations are restated here.

Estimation of ingot temperatures

The Kalman filter produces the state estimation and the one-step ahead prediction, given by

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{k}(k)(y(k) - \mathbf{c}^T\hat{\mathbf{x}}(k|k-1)), \qquad (6.8.3)$$

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k) + \mathbf{b}u(k). \qquad (6.8.4)$$

k is the Kalman gain vector. Thus at any instant k, the estimates of the ingot-surface and the ingot-centre temperatures are obtained from (6.8.3).

Ready-to-roll time prediction

Here, two basic pieces of information are required:
(i) the specification of the ready-to-roll time, and
(ii) the future control input (i.e. the fuel flow) sequences or the model for the controller, which is required for the temperature prediction.

The ready-to-roll time is often defined in terms of the temperature windows within which the ingot-centre temperature and the ingot-surface temperature must lie. These values may be typically 1280°±40°C and 1280°±20°C respectively. The temperature ranges may vary depending on the type of steel and the rolling practice.

The p-step prediction of the ingot temperatures can be obtained from the optimal state predictor (6.7.5),

$$\hat{x}(k+p|k) = A^p\hat{x}(k|k) + \sum_{i=k}^{m} A^{m-i}bu(i), \qquad (6.8.5)$$

where m = k+p-1. Alternatively use

$$\hat{x}(k+p|k) = A\hat{x}(k+p-1|k) + bu(k+p-1). \qquad (6.8.6)$$

The prediction equations (6.8.5) and (6.8.6) use the future values of u, which can be computed as follows. Often a PI (proportional + integral) controller is used for the pit temperature control; the future control input sequences can be obtained from

$$u(i+1) = c_p\varepsilon(i+1) + c_q\sum_{j=i+1-N}^{i+1} \varepsilon(j), \qquad (6.8.7)$$

where

$$\varepsilon(i+1) = y_{set}(i+1) - \hat{y}(i+1|i),$$

$$\hat{y}(i+1|i) = c^T\hat{x}(i+1|i).$$

ε is the deviation of the pit wall temperature from the desired set point y_{set}; the sequence $\{y_{set}(k)\}$ is assumed to be known. $\Sigma\varepsilon$ is the integrated value of ε over the appropriate (past) horizon N, c_p and c_q are the proportional and the integral constants of the PI controller.

Thus starting from p=1, sequential multistep prediction of the state can be computed using (6.8.6–6.8.7) to determine the predicted ready-to-roll time.

Summary

(1) Formulate the state-space model (6.8.1–6.8.2) for the soaking pit and the ingot heating process.

(2) Specify the initial condition, and use a suitable heat transfer model to generate the ingot-surface and the ingot-centre temperatures against the progression of time.

(3) Use the ingot temperatures so obtained from the heat transfer model in the state-space description (6.8.1),and estimate the parameters of the state-space model using the least squares method.

(4) Define the ready-to-roll time in terms of the desired ingot temperatures. Define the parameters of the pit temperature controller.

(5) Run the sequential multistep prediction algorithm (6.8.6) to predict the ready-to-roll time.
(6) From the state estimates (6.8.3), indicate when the ready-to-roll time is actually reached.

Remarks
(1) If the predicted ready-to-roll time is too long ahead of the mill availability, the heating strategy can be suitably modified to suit the needs of the mill. Again, prediction of the ready-to-roll time well in advance helps schedule the operation of the battery of pits.
(2) The time when the pit will go into in-control phase, can be predicted the same way as the ready-to-roll time.
(3) In spite of large variability in the ingot soaking process, the presented estimation and prediction scheme should improve the operational consistency as well as the fuel efficiency. The standard deviation of the average ingot temperature is also expected to decrease.

6.9 CONCLUDING REMARKS

In this chapter, state-space modelling and optimal estimation of states using the Kalman filter were studied. Compared with the transfer function models based on input-output data, the state-space approach offers the additional flexibility of accommodating internal variables as states, which may not be accessible or measurable. The state-space approach permits use of a large number of widely studied and well-established methods and algorithms for estimation, prediction, smoothing and control.

The Kalman filter can produce optimal state estimates under steady state conditions, even when the measurements are noisy, conditional on the noise being independent with Gaussian distribution. In practical applications, there are two basic problems: (i) the parameters of the state-space model may be unknown or slowly time-varying, and (ii) the noise may not be truly independent and Gaussian. So the optimality of the estimates will suffer. Nevertheless, the Kalman filter can produce workably good results and hence is widely used for real-life applications.

The Kalman filter itself produces one-step ahead prediction; multistep predictions are obtained through repeated use of shorter-step predictions.

REFERENCES

Remarks: State-space modelling is discussed in many texts including [5,12,24]. Optimum estimation including Bayesian estimation is studied in [21]. Kalman filtering, introduced in [10], is discussed in the context of optimal estimation in [4,15,16,21]. Detailed study of Kalman filtering, along with extended the Kalman filtering, also appears in [1,9]. Adaptive Kalman filtering is discussed in [17,19]. Many prediction applications of Kalman filter have been reported, for example [18,22]; an application to the modelling of a periodic time-series appears in [25]. The structural modelling in state-space framework is discussed in [7,20, 27]. The implementational aspects of the Kalman filter are discussed in [3,6]. The state-space modelling and temperature prediction of soaking pits using the Kalman filter are studied in [13,26]. Important theoretical and application papers on Kalman filtering are collected in [23].

[1] Anderson, B.D.O., and J. B. Moore (1979): *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, N.J.

[2] Åström, K.J., and B. Wittenmark (1973): 'On self-tuning regulators', *Automatica*, 9, 185-199.

[3] Bierman, G. J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[4] Bryson, A.E., and Y.C. Ho (1969): *Applied Optimal Control*, Ginn (Blaisdell), Walthlam, Mass.

[5] Cadzow, J.A., and H.R. Martens (1970): *Discrete-time and Computer Control Systems*, Prentice-Hall, Englewood Chiffs, N.J.

[6] Grewal, M.S., and A.P. Andrews (1993): *Kalman Filtering: Theory and Practice*, Prentice Hall, Englewood Cliffs, New Jersey.

[7] Harvey, A.C. (1990): *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge Univ. Press, Cambridge.

[8] Hinami, M., S. Konoeda, T. Oiwa, and A. Inni (1975): 'Development of a computerized system for predicting the progress of soaking in the soaking pit', *The Sumitomo Search*, 13, May, 1-7.

[9] Jazwinski, A. H. (1970): *Stochastic Processes and Filtering Theory*, Academic Press, New York.

[10] Kalman, R.E. (1960): 'A new approach to linear filtering and prediction problems', *Trans. ASME*,

Journal of Basic Engineering, 82 D, 35–45.

[11] Kung, E. Y., J. R. Dahn, and G. B. Delancy (1967): 'A mathematical model of soaking pits', *ISA Trans.*, 6, 162–168.

[12] Kuo, B.C. (1980): *Digital Control Systems*, Holt, Rinehart and Winston, Inc., New York.

[13] Lumelsky, V.J. (1983): 'Estimation and prediction of unmeasurable vairables in the steel mill soaking pit control system', *IEEE Trans. on Automatic Control*, AC-28, No.3, 388–400.

[14] Massey, I.D., and A.T. Sheridan (1971): 'Theoretical prediction of earliest rolling times and solidification times of ingots', *Journal of Iron and Steel Inst.*, 209(5), 391–395.

[15] Maybeck, P. S. (1982): *Stochastic Models Estimation and Control*, Vol.2, Academic Press, New York.

[16] Meditch, J. S. (1979): *Stochastic Optional Linear Estimation and Control*, McGraw-Hill, New York.

[17] Mehra, R.K. (1972): 'Approaches to adaptive filtering', *IEEE Trans. on Auto. Control*, AC-17(5), 693–698.

[18] Mehra, R.K. (1979): 'Kalman Filters and their Applications to Forecasting', *Management Sciences* (Spl. Issue: *Forecasting*), Reprinted in TIMS Studies in Management Sciences, Eds. S. Makridakis and S.C. Wheelwright, Vol.12, North Holland, Amsterdam, 75–94.

[19] Moghaddamjoo, A. (1989): 'Approaches to adaptive Kalman Filtering', *Control Theory and Advances Technology*, 5(1), 1–18.

[20] Pagan, A. (1975): 'A note on extraction of components from a time series', *Econometrica*, 43, 163–168.

[21] Sage, A.P. and J. L. Melsa (1971): *Estimation Theory with Applications to Communications and Control*, McGraw-Hill, New York.

[22] Sastri, T. (1985): 'A state-space modelling approach for time series forecasting', *Management Science*, 31(11), 1451–1470.

[23] Sorenson, H.W. (Ed.) (1985): *Kalman Filtering: Theory and Applications*, IEEE Press, New York.

[24] Strejc, V. (1981): *State Space Theory of Discrete Linear Control*, Wiley, Chichester.

[25] Szelag, C.R. (1982): 'A Short-term Forecasting Algorithm for Trunk Demand Servicing', *The Bell System Technical Journal*, 61(1), 67–77.

[26] Wick, H.J. (1982): 'On-line estimation of temperature

profiles in steel ingots in a soaking pit', *Journal A*, **23**(2), 61–68.

[27] Young, P. (1988): 'Recursive extrapolation, interpolation and smoothing of nonstationary time-series', in *Identification and System Parameter Estimation*, Ed. H.F. Chen., Pergamon Press, Oxford.

CHAPTER 7

ORTHOGONAL TRANSFORMATION AND MODELLING OF PERIODIC SERIES

Two basic consequences of orthogonal transformation are relative decorrelation of data and compression of information, which can be used for modelling and prediction of periodic series.

7.1 INTRODUCTION

The theory and application of orthogonal transformation is discussed in this chapter; the study concentrates on the singular value decomposition and the Walsh-Hadamard transformation.

Orthogonal transformation converts mutually correlated set of data into a relatively decorrelated set of transform coefficients (or spectral components). The energy in the data, which represents the information content, remains conserved in the transformation process but the distribution of energy becomes more compact following transformation compared with the original time domain, where the energy distribution is rather uniform. The transformation process is linear and reversible.

Two closely related formats of orthogonal transformation studied in this chapter are as follows:

(i) The transform **y** of the data vector **x** is obtained by multiplying **x** by the orthogonal matrix **W**:

$$\mathbf{y} = \mathbf{W}\mathbf{x}.$$

Karhunen-Loève transformation, Walsh-Hadamand transformation etc. are based on this format.

(ii) The data or information matrix is decomposed into orthogonal factors. Singular Value Decomposition (SVD) has such a format. For example, SVD of a matrix **X** is given by

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{T},$$

where **U** and **V** are orthogonal matrices, and **S** is a diagonal matrix of the singular values **X**.

Orthogonal transformation results in relative decorrelation of the data and compaction of information. Hence it is advantageous to perform modelling and prediction operations with the transformed data; the time domain results can be obtained through reverse transformation.

Orthogonal transformation offers some of the strongest tools for data analysis, modelling, prediction and filtering. It is applicable to periodic or aperiodic processes, linear or nonlinear processes, simple single-input single-output or complex multi-input multi-output processes. This chapter concerns the modelling of data series which are nearly periodic in nature; applications to quasiperiodic series and complex multivariables processes are studied in Chapters 10 and 11.

This chapter opens with an introduction to the basics of orthogonal transformation in Sec.7.2, where the characteristic features and the properties are discussed. Next the theory of Karhunen-Loève transformation (KLT) is studied in Sec.7.3. KLT is an eigenvalue based optimal orthogonal transformation; although the application of KLT is not included in the modelling and prediction methods discussed in this chapter, it is expected that a study of KLT can help the understanding of orthogonal transformation.

The two types of transformations studied in this chapter are the Walsh Hadamard Transformation (WHT) and the Singular Value Decomposition (SVD). WHT is extremely simple in implementation, whereas SVD is one of the most numerically robust algebraic operations. SVD features widely in this book both as a tool for matrix operations as well as an integral part of algorithms designed for the analysis of data and modelling of processes. Both WHT and SVD are noncircular methods of orthogonal transformation; among the popular circular methods are Fast Fourier transform (FFT) (discussed in Chapter 2), and Discrete cosine transform (DCT).

The theoretical aspects of WHT are presented in Sec.7.4, and the applications of WHT to some real-life prediction problems are discussed in Sec.7.5. The theory of Singular Value Decomposition is studied along with its characteristic features in Sec.7.6. In Sec.7.7, the characterization of a periodic process in terms of the decomposition components of SVD is presented, which is central to the applications of SVD for the modelling and prediction of periodic time series discussed in Sec.7.8.

This chapter is supported mainly by six appendices;

Appendix 7A presents details of the solutions of some of the examples featuring in this chapter. Appendix 7B to 7F contain some real-life time series data used in this book.

7.2 BASICS OF ORTHOGONAL TRANSFORMATION

Orthogonality and orthonormality

Normal functions: A function is said to be normal if its inner (or scalar) product with itself is unity.

Orthogonal functions or signals: Two signals $\{y(k)\}$ and $\{z(k)\}$ are said to be orthogonal over an interval $k = 1$ to m, if

$$\sum_{k=1}^{m} y(k)z(k) = \begin{cases} 0 \text{ for } m \neq n, \\ C \text{ for } m = n, \end{cases} \quad C \text{ being a constant.}$$

Orthogonal vectors: Two vectors $\mathbf{x}$ and $\mathbf{y}$ are orthogonal if their scalar product is zero, i.e. $\mathbf{x}^T\mathbf{y} = 0$. The n-vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$ are mutually orthogonal, if they are pairwise orthogonal.

Orthonormal vectors: The n-vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$ are orthonormal, if

(a) they are pairwise orthogonal:

$$\mathbf{v}_i^T\mathbf{v}_j = 0, \qquad i \neq j \text{ for } i, j = 1, \dots, r, \qquad \text{and}$$

(b) they are individually normalized:

$$\mathbf{v}_i^T\mathbf{v}_i = 1 \qquad i = 1, \dots, r.$$

Such a set of vectors is called an *orthonormal set.*

Orthogonal matrix

A square matrix $\mathbf{V}$ is orthogonal, if and only if

$$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}. \tag{7.2.1}$$

It is implied that both the row vectors and the column vectors of the square matrix $\mathbf{V}$:

$$\mathbf{V} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{n1} & V_{n2} & \cdots & V_{nn} \end{bmatrix},$$

form normal orthogonal systems, that is

$$\mathbf{v}_k^T\mathbf{v}_j = \sum_{i=1}^{n} v_{ik}v_{ij}, \qquad k,j = 1,2,\ldots,n,$$

$$= \delta_{kj}, \qquad \delta_{kj} \begin{cases} = 1, \text{ if } k=j, \\ = 0, \text{ otherwise.} \end{cases}$$

An orthogonal matrix has a determinant of +1 or −1 but the converse is not true.

Orthogonal transformation

For any vector **x**, the linear transformation

$$\mathbf{y} = \mathbf{Ax}$$

is an orthogonal transformation, if the absolute value (or the Euclidean length) of **x** is preserved through the transformation, that is

$$|\mathbf{y}| = |\mathbf{Ax}| = |\mathbf{x}|,$$

where the transformation matrix **A** is an orthogonal matrix:

$$\mathbf{A}^T\mathbf{A} = \mathbf{AA}^T = \mathbf{I},$$

whereas multiplication by a general nonsingular matrix may alter the absolute magnitude of a vector drastically. In other words, the 2-norm of a vector **x** remains unchanged under orthogonal transformation,

$$\|\mathbf{Ax}\|_2^2 = \mathbf{x}^T\mathbf{A}^T\mathbf{Ax},$$
$$= \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|_2^2.$$

Similarly, for any orthogonal matrices **A** and **B** of appropriate dimensions, the matrix 2-norm and the Frobenius norm of any mxn matrix **Y** remain invariant under orthogonal transformation:

$$\|\mathbf{AYB}\|_2 = \|\mathbf{Y}\|_2,$$
$$\|\mathbf{AYB}\|_F = \|\mathbf{Y}\|_F.$$

General properties

(a) Uniqueness: Orthogonal signals or vectors carry discriminating information and are maximally independent.

(b) Energy conservation and compaction: The total energy in the transformed data is the same as in the time or spatial domain (that is as before transformation), however while the energy is usually uniformly distributed in the time or spatial domain, the orthogonal tranformation results in

concentration of energy to fewer points in the transformed data. This feature of compaction of energy can be used for data compression, since the relatively insignificant data elements can be discarded without any conceivable loss of information content.

(c) Noise immunity: Operations with the transformed data enjoy excellent noise immunity. Perturbations in the original data are carried usually at much attenuated level in the transformed data; in any case the perturbations cannot be larger than that in the original data.

(d) Linearity and reversibility: Orthogonal transformation is a linear and reversible transformation but all linear or reversible transformations are not orthogonal.

(e) Physical interpretation: The physical sense in the data is apparently destroyed by orthogonal transformation but the transformed data are of greater computational and statistical significance. No information is lost through the transformation, as the transformation is reversible, and the original data can be reconstructed by reverse transformation.

(f) Causality: As against a transfer function model or time series model, which has a causal structure, no causality is presumed in orthogonal transform representation.

7.3 KARHUNEN-LOÈVE TRANSFORM

Karhunen-Loève expansion or transformation (KLT) can produce optimal representation of the information in the data, in reduced dimension; the optimality is in minimising the mean square transformation error. The basic idea is to map the data vector into the transform components through orthogonal transformation; the optimal choice of these components will lead to the optimal transformation, which on reverse transformation should produce minimum mean square error reconstruction of the original data.

KLT lacks popularity because of its high computational requirement. However, the study of KLT is useful in understanding the *optimal transformation* aspect, which is unique in orthogonal transformation.

The problem

Let $\mathbf{x}$, the observed data vector, be mapped onto $\mathbf{y}$ through the transformation:

$$\mathbf{x} = \mathbf{W}\mathbf{y} = \sum_{i=1}^{n} y_i \mathbf{w}_i, \qquad (7.3.1)$$

where

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T, \qquad \mathbf{y} = [y_1, y_2, \ldots, y_n]^T,$$

and $\mathbf{W}$ is an nxn orthogonal matrix:

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n], \qquad \mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}.$$

Hence

$$y_i = \mathbf{w}_i^T \mathbf{x}.$$

Each component y_i in the transform domain represents a discriminating feature of $\mathbf{x}$. $\mathbf{w}_i$ which represent n-dimensional transform space are called the *basis vectors.*

If $\mathbf{y}_m$, a subset of $\mathbf{y}$ or a reconfigured .$\mathbf{y}$, is used to represent $\mathbf{x}$, then the transformation will be *optimal* if the mean square error $\|\mathbf{x}-\mathbf{W}\mathbf{y}_m\|^2$ is minimized.

The conditions under which the transformation is optimal are discussed.

Optimal transformation

Consider the transform vector

$$\mathbf{y}_m = [y_1, y_2, \ldots, y_m, b_{m+1}, b_{m+2}, \ldots, b_n]^T,$$

where $b_{m+1}, b_{m+2}, \ldots, b_n$ are preselected constants. Hence the estimated data vector is

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y}_m.$$

The estimation error is given by

$$\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}} = \sum_{i=m+1}^{n} (y_i - b_i)\mathbf{w}_i; \qquad (7.3.2)$$

the mean square error is given by

$$J = E\|\tilde{\mathbf{x}}\|^2. \qquad (7.3.3)$$

J is minimized for the optimum values of b_i and $\mathbf{w}_i$. It can be shown that following (7.3.2-7.3.3) the value of b_i that minimizes J is given by

$$b_i = \mathbf{w}_i^T\bar{\mathbf{x}}, \qquad (7.3.4)$$

where $\bar{\mathbf{x}} = E(\mathbf{x})$. For this value of b_i following (7.3.4, 7.3.3 and 7.3.2) J becomes

$$J = \sum_{i=m+1}^{n} \mathbf{w}_i^T \mathbf{P}_x \mathbf{w}_i,$$

where

$$\mathbf{P_x} = E\{(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\}$$

is the covariance matrix of **x**. The optimum value of $\mathbf{w}_i$ is given by

$$\mathbf{P_x w}_i = k_i \mathbf{w}_i,$$

which by definition implies that k_i are the eigenvalues λ_i, and $\mathbf{w}_i$ are the corresponding eigenvectors of the covariance matrix $\mathbf{P_x}$. Hence the minimum mean square error is given by

$$J_{min} = \sum_{i=m+1}^{n} \lambda_i. \qquad (7.3.5)$$

The conditions for the optimal transform are derived in Ahmed and Rao (1975, p.200).

The expansion of the data vector **x** in terms of the eigenvectors of its covariance matrix: $\mathbf{x} = \mathbf{Wy}$, as in (7.3.1), is referred to as the Karhunen-Loève expansion, and $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ is called the Karhunen-Loève transform.

Remark: In statistics, Principal Component Analysis (PCA) refers to the optimal linear transformation $\mathbf{y} = \mathbf{W}^T\mathbf{x}$. Again, Hotelling transform in Communication theory has the same structure.

Characteristic features

(a) Uncorrelated features: Since $\mathbf{y} = \mathbf{W}^T\mathbf{x}$, the covariance of **y** is diagonal:

$$\begin{aligned}\mathbf{P_y} &= \mathbf{W}^T\mathbf{P_x W}\\ &= \text{Diag. } [\lambda_1, \lambda_2, \ldots, \lambda_n].\end{aligned}$$

Hence the feature elements y_i are mutually uncorrelated. If $\{x_i\}$ have normal distribution, the features $\{y_i\}$ will be mutually independent.

(b) Optimal feature selection and ordering: The relative richness of information contained in the transform elements y_i depends on the relative magnitude of the corresponding eigenvalues λ_i. If the eigenvalues are arranged in order of nonincreasing magnitude:

$$\lambda_1 > \lambda_2 > \lambda_3 > \ldots > \lambda_n > 0,$$

then the ordering of the features y_1, y_2, y_3, ... will conform to a descending degree of importance. This reordered format is called the *Generalized Karhunen-Loève transform.*

If n-dimensional x is now represented by r significant transformed elements as

$$y' = [y_1,\ldots, y_r]^T, \qquad r \ll n,$$

$$J = \sum_{i=r+1}^{n} \lambda_i,$$

where $\lambda_{r+1},\ldots, \lambda_n$ are the lowest (n-r) eigenvalues and hence y_r will provide minimum mean square error feature measurement of y. Thus KLT offers a procedure for feature selection and ordering which is statistically optimal.

(c) Computational aspects: Although KLT is an optimal transformation, its main drawback is the high computational requirement which is of the order of $2n^2$ multiplications. There being no fast algorithms available for KLT, often suboptimal transforms, e.g., Fast Fourier Transform, Walsh Transform, Discrete Cosine Transform etc. are used for which fast algorithms are available.

7.4 WALSH-HADAMARD TRANSFORM

The Walsh-Hadamard transform is one of the most widely used nonsinusoidal, suboptimal orthogonal transforms. While Fourier transform decomposes signals or data sequences into sinusoidal components, Walsh transform decomposes the same into rectangular pulse sequences called *Walsh functions*. As proposed by J.L. Walsh in 1923, the Walsh functions form a complete set of orthogonal rectangular pulses of magnitude +1 or -1.

Fig.7.4.1 shows a set of eight Walsh functions, where the following structural characteristics may be noted:

(a) A Walsh function is completely specified by the time period t and an ordering index i:

$$\{wal(i,t),\ i = 0, 1,\ldots, n-1\},$$

where $n = 2^p$, p being a positive integer; n number of functions constitute a particular set.

(b) The time period is divided into n equal subintervals. The function changes sign only where t is a multiple of a power of 1/2.

(c) The functions are arranged in increasing order of zero-crossings. The number of zero-crossings within a unit time period is called *sequency*.

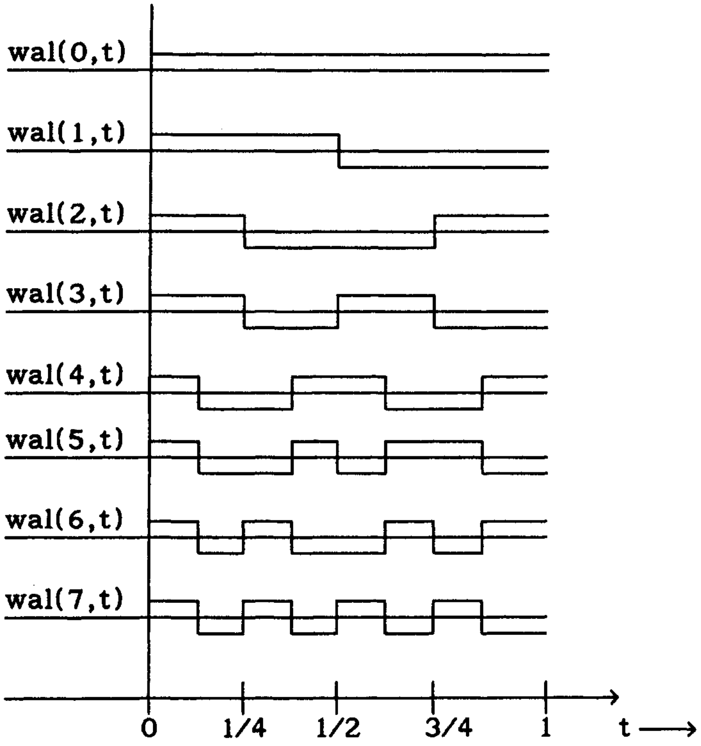


Figure 7.4.1 Walsh ordered continuous-time Walsh functions, n = 8.

$$\mathbf{H}_n = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \hdashline 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ \hdashline 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ \hdashline 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{array}\right] \begin{array}{c} \textit{Sequency} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

Figure 7.4.2 Matrix representation of Walsh ordered discrete-time Walsh functions, n = 8.

If the continuous-time Walsh functions in an unit interval, as shown in Fig.7.4.1, are sampled at n equidistant sub-intervals, an equivalent discrete-time representation in terms of an nxn matrix results (Fig.7.4.2). This matrix $\mathbf{H}_n$, is a subset of a class of matrices called *Hadamard matrices.* Because of the representation of the Walsh functions in terms of Hadamard matrices, the corresponding orthogonal transformation is also called Walsh-Hadamard transformation (WHT). The sampled sets of Walsh functions, given by the rows of $\mathbf{H}_n$, are called the *basis functions.*

The discrete Walsh functions are

(a) mutually orthogonal:

$$\sum_{t=0}^{n} \mathrm{wal}(i,t)\mathrm{wal}(j,t) = \begin{cases} 0, \text{ for } i \neq j, \\ n, \text{ for } i = j, \end{cases}$$

and

(b) symmetric: wal(i,t) = wal(t,i).

7.4.1 Generation of Walsh Functions using Hadamard Matrices

There are various procedures for the generation of the Walsh functions by software and hardware means (Beauchamp, 1984, Alexandridis, 1989). One of the most efficient methods of performing WHT is by using Hadamard matrices.

The Hadamard matrix **H**, originally introduced by French mathematician M.J. Hadamard in 1893, is an nxn square matrix of +1 and -1 elements with the following features:

(a) $\mathbf{H}^T\mathbf{H} = n\mathbf{I}$, **I** being the identity matrix

(b) $n = 2^p$, p being a positive integer

(c) **H** can be reconfigured such that all elements of the 1st row and the 1st column are positive, and both sum up to n.

The lowest order Hadamard matrix is of order 2:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The higher order matrices can be generated recursively as

$$\mathbf{H}_n = \begin{bmatrix} \mathbf{H}_{n/2} & \mathbf{H}_{n/2} \\ \mathbf{H}_{n/2} & -\mathbf{H}_{n/2} \end{bmatrix}, \tag{7.4.1}$$

where $n = 2^p$, $n/2 = 2^{p-1}$, and p = 2,3,4 etc. For example, $\mathbf{H}_8$ can be expressed as shown in Fig.7.4.3. The recursive relation given by (7.4.1) may also be expressed as

$$H_n = H_{n/2} \otimes H_2, \tag{7.4.2}$$

where $\otimes$ stands for direct or Kronecker product which implies substitution of each element of $H_{n/2}$ by H_2 in (7.4.1).

Note that the rows of the Hadamard matrix Fig.7.4.3, generated by the recursive relationship (7.4.1), represent the same eight sampled set of Walsh functions as in Fig.7.4.2; however the rows in Fig.7.4.3 are said to be *natural ordered* or *Kronecker ordered* unlike those of Fig.7.4.2 which are *sequency ordered.*

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \left[\begin{array}{cc:cc} H_2 & H_2 & H_2 & H_2 \\ H_2 & -H_2 & H_2 & -H_2 \\ \hdashline H_2 & H_2 & -H_2 & -H_2 \\ H_2 & -H_2 & -H_2 & H_2 \end{array}\right]; \quad \text{or}$$

$$H_8 = \begin{array}{c} \\ \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \hdashline 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hdashline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ \hdashline 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right] \end{array} \quad \begin{array}{c} \textit{Sequency} \\ 0 \\ 7 \\ 3 \\ 4 \\ 1 \\ 6 \\ 2 \\ 5 \end{array}$$

Figure 7.4.3 Natural or Kronecker ordered nxn Hadamard matrix H_n, n = 8.

There are fast algorithms available both for sequency ordered as well as natural ordered Walsh Hadamard transforms. The total number of additions and subtractions required for implementation is $n\log_2 n$.

A summary of the basic properties of Hadamard matrices used in WHT follows.

(a) The Hadamard matrix H_n is an nxn square symmetric matrix of +1 and −1 elements, where $n = 2^p$, p being a positive integer.

(b) $H_n^T H_n = nI_n$ and $H_n^{-1} = \frac{1}{n}H_n$. (7.4.3)

(c) All rows (as well as columns) of H_n add up to an n vector: [n 0 0 ... 0].

(d) A matrix H, generated by Kronecker product of two

Hadamard matrix H_m and H_n, is a symmetric matrix of order mn:

$$\begin{aligned} H &= H_m \otimes H_n \\ &= H_{mn} = mnI_{mn}. \end{aligned}$$

7.4.2 One-Dimensional WHT

The Walsh-Hadamard transform of an n-data vector x is given by

$$x_w = \frac{1}{n}H_n x, \qquad n = 2^p,\ p = 1,2,3,\ldots, \tag{7.4.4}$$

where H_n is nxn Hadamard matrix, and x_w is the transformed vector. Note that if the data set is of some arbitrary length it has to be either truncated or extended with dummy data, such that it is n-long.

Using (7.4.3), the reverse transform from Walsh domain to time or spatial domain is obtained as

$$x = H_n x_w. \tag{7.4.5}$$

Example

If $x = [2 \ \ -1 \ \ 3 \ \ 5]^T$, n = 4,

$$x_w = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \\ 3 \\ 5 \end{bmatrix},$$

or

$$x_w = [9/4 \ \ 1/4 \ \ -7/4 \ \ 5/4]^T.$$

Characteristic features

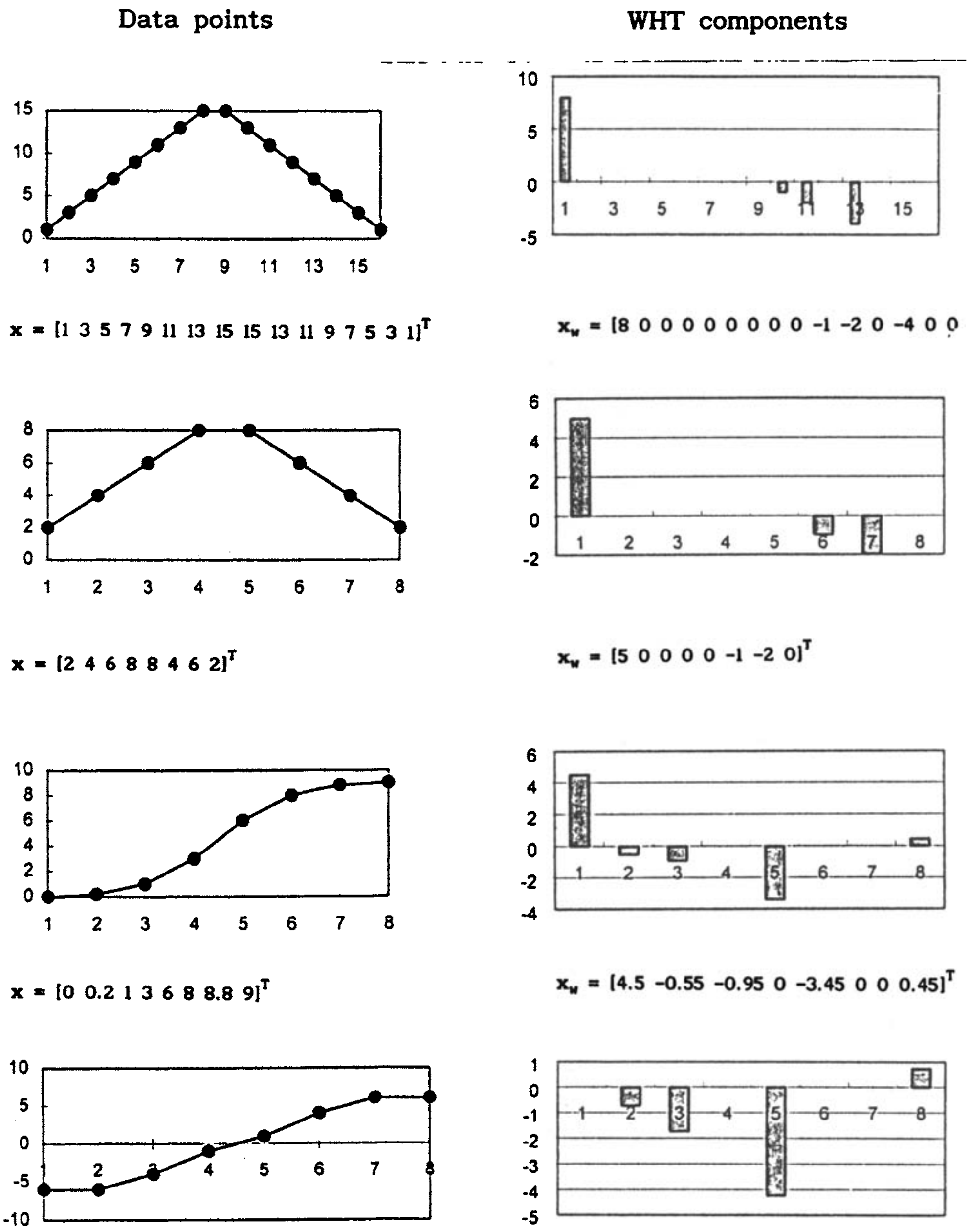
WHT of some typical data patterns are shown in Fig.7.4.4; the following features may be noted.

(a) $x_w(1)$ gives the average value of the signal. It is the same as the zero-frequency term of the trigonometric Fourier series. $x_w(1) = 0$ implies the data vector being zero mean.

(b) $x_w(2)$ is indicative of asymmetry about the centre; its small magnitude will mean the signal will be relatively symmetric about the centre line.

(c) For a strictly symmetric signal of length 2^p, p = 1, 2, etc., the number of prime components will be p with the rest 2^p-p terms being zero.

(d) The energy in x_w is n times that of x, where n = 2^p.



Data points

WHT components

$\mathbf{x} = [1\ 3\ 5\ 7\ 9\ 11\ 13\ 15\ 15\ 13\ 11\ 9\ 7\ 5\ 3\ 1]^T$

$\mathbf{x}_W = [8\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -1\ -2\ 0\ -4\ 0\ 0$

$\mathbf{x} = [2\ 4\ 6\ 8\ 8\ 4\ 6\ 2]^T$

$\mathbf{x}_W = [5\ 0\ 0\ 0\ 0\ -1\ -2\ 0]^T$

$\mathbf{x} = [0\ 0.2\ 1\ 3\ 6\ 8\ 8.8\ 9]^T$

$\mathbf{x}_W = [4.5\ -0.55\ -0.95\ 0\ -3.45\ 0\ 0\ 0.45]^T$

$\mathbf{x} = [-6\ -6\ -4\ -1\ 1\ 4\ 6\ 6]^T$

$\mathbf{x}_W = [0\ -0.75\ -1.75\ 0\ -4.25\ 0\ 0\ 0.75]^T$

Figure 7.4.4 One-dimensional Walsh-Hadamard transform of some typical data patterns.

Remark: Implementation of WHT
Since WHT involves only additions, subtractions and division by 2^p, the computational load is minimal; WHT can also be easily hardware implemented.

Example 7.4.2 WHT of the atmospheric CO_2 series

Consider the monthly data for 10 years for the atmospheric CO_2 variations given in Appendix 7C. The data for each year are arranged into the rows of a matrix **A**; each row of **A** is appended with 4 zero elements to make the row length equal to 2^4. **A** and its WHT matrix $\mathbf{A}_w$ are as follows.

$$\mathbf{A} =$$

$$\begin{bmatrix}
315.16 & 315.97 & 316.37 & 317.40 & 317.96 & 317.82 & 316.23 & 314.54 & 313.60 & 313.03 & 314.57 & 315.32 & 0 & 0 & 0 & 0 \\
316.10 & 316.68 & 317.37 & 318.79 & 319.63 & 319.29 & 317.86 & 315.55 & 313.85 & 313.64 & 314.61 & 315.81 & 0 & 0 & 0 & 0 \\
316.54 & 317.34 & 318.12 & 319.06 & 320.20 & 319.44 & 318.24 & 316.52 & 314.57 & 315.13 & 315.75 & 316.73 & 0 & 0 & 0 & 0 \\
317.70 & 318.29 & 319.37 & 320.25 & 320.84 & 320.43 & 319.35 & 317.13 & 316.01 & 315.19 & 316.42 & 317.47 & 0 & 0 & 0 & 0 \\
318.45 & 318.82 & 319.72 & 321.06 & 321.87 & 321.22 & 319.44 & 317.48 & 315.89 & 315.83 & 316.72 & 317.98 & 0 & 0 & 0 & 0 \\
319.32 & 320.36 & 320.82 & 322.06 & 322.17 & 321.95 & 321.20 & 318.81 & 317.82 & 317.37 & 318.93 & 319.09 & 0 & 0 & 0 & 0 \\
319.94 & 320.98 & 321.81 & 323.03 & 323.36 & 323.11 & 321.65 & 319.64 & 317.86 & 317.25 & 319.06 & 320.26 & 0 & 0 & 0 & 0 \\
321.65 & 321.81 & 322.36 & 323.67 & 324.17 & 323.39 & 321.93 & 320.29 & 318.58 & 318.60 & 319.98 & 321.25 & 0 & 0 & 0 & 0 \\
321.88 & 322.47 & 323.17 & 324.23 & 324.88 & 324.75 & 323.47 & 321.34 & 319.56 & 319.45 & 320.45 & 321.92 & 0 & 0 & 0 & 0 \\
323.40 & 324.21 & 325.33 & 326.31 & 327.01 & 326.24 & 325.37 & 323.12 & 321.85 & 321.31 & 322.31 & 323.72 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\mathbf{A}_w^T = \frac{1}{16}\,\mathbf{W}\mathbf{A}^T; \qquad \mathbf{A}_w = \frac{1}{16} \times$$

$$\begin{bmatrix}
3788.0 & -0.2 & -0.9 & -0.0 & 1254.9 & -3.8 & -10.9 & 3.1 & 1274.9 & 0.2 & 5.6 & -2.7 & -1258.2 & -3.5 & -4.4 & 0.4 \\
3799.2 & -0.3 & -0.8 & 0.3 & 1254.5 & -5.6 & -11.8 & 4.2 & 1283.4 & 1.6 & 5.1 & -2.5 & -1261.3 & -3.7 & -6.0 & 1.4 \\
3807.6 & -0.8 & -1.2 & -0.4 & 1258.8 & -5.8 & -11.0 & 1.5 & 1283.3 & 2.3 & 4.4 & -1.2 & -1265.5 & -2.7 & -5.4 & 0.7 \\
3818.5 & 0.9 & -1.5 & 0.3 & 1263.0 & -4.3 & -11.1 & 4.0 & 1288.3 & 1.4 & 3.9 & -3.4 & -1267.2 & -3.9 & -5.7 & 0.2 \\
3824.5 & -0.3 & -0.3 & 1.0 & 1264.5 & -5.5 & -12.7 & 3.6 & 1291.6 & 2.1 & 5.6 & -1.7 & -1268.4 & -3.1 & -6.7 & 1.0 \\
3839.9 & 0.6 & -1.9 & -1.4 & 1271.6 & -4.6 & -10.1 & 3.0 & 1293.5 & 0.0 & 3.7 & -2.6 & -1274.8 & -5.2 & -4.5 & 1.8 \\
3848.0 & -0.6 & -2.9 & 0.2 & 1272.4 & -5.1 & -13.3 & 3.7 & 1299.1 & 0.6 & 5.5 & -3.4 & -1276.4 & -3.9 & -4.9 & 0.1 \\
3857.7 & -0.3 & -1.3 & 1.5 & 1278.1 & -5.2 & -12.0 & 3.3 & 1300.9 & 2.2 & 6.8 & -1.0 & -1278.7 & -2.6 & -3.9 & 0.8 \\
3867.6 & -0.8 & -1.6 & 0.0 & 1278.7 & -5.3 & -11.2 & 4.1 & 1304.8 & 2.0 & 5.1 & -3.1 & -1284.1 & -2.6 & -4.5 & 0.9 \\
3890.2 & 0.4 & -2.1 & 0.6 & 1286.7 & -5.7 & -11.7 & 3.6 & 1311.8 & 2.1 & 3.6 & -3.3 & -1291.7 & -3.9 & -5.9 & -0.3
\end{bmatrix}$$

Note that only some of the columns of $\mathbf{A}_w$ (that is columns 1, 5, 9, and 13) carry the major part of the information. The sum of squares of the elements of individual columns (1 to 16) are found to be as follows:

574268.202, 0.013, 0.103, 0.024,
62841.724, 1.028, 5.264, 0.474,
65326.371, 0.109, 0.987, 0.268,
63268.571, 0.502, 1.078, 0.032.

The implication is that **A** can be reconstructed with little loss of information using only 4 columns of A_w. This feature is used in the prediction of periodic series as discussed in Sec.7.5.

7.4.3 Two-Dimensional WHT

Often the data set arranged in the form of a matrix may have correlation both in the direction of rows and columns which justifies two-dimensional Walsh-Hadamard transformation:

$$X_w = \frac{1}{n_1 n_2} H_{n_1} X H_{n_2},$$

where X is $n_1 x n_2$ data matrix, X_w is its transform, and $n_1 = 2^{p_1}$, $n_2 = 2^{p_2}$, p_1 and p_2 being positive integers. The reverse transformation is given by

$$X = H_{n_1} X_w H_{n_2}.$$

If $n_1 = n_2$,

$$X_w = (1/n^2) H_n X H_n \quad \text{and} \quad X = H_n X_w H_n.$$

Since $H_n^{-1} = \frac{1}{n} H_n$, X_w may also be expressed as

$$nX_w = H_n^{-1} X H_n,$$

which is in similarity transformation format, $[nX_w]$ and X being similar matrices.

Remark: Similarity transformation
Two square matrices X and Y are *similar*, if there is a nonsingular matrix G such that $Y = G^{-1}XG$; G is called the similarity transformation matrix. If X and Y are similar, their eigenvalues, characteristic polynomials and determinants are equal. If v is the eigenvector of X, $G^{-1}v$ will be the eignvector of Y. Y may not be a diagonal matrix. A square matrix X is called *defective*, if it is not diagonable by similarity transformation.

□□

The two-dimensional WHT is applicable for data compression where the information is contained in two-dimensional space; such applications for image processing are discussed in Alexandridis (1989).

7.5 PREDICTION BASED ON WHT

7.5.1 Basic Principle

The basic idea is to extrapolate the data sequence following transformation, and to perform reverse transformation to produce prediction.

Let X be the mxn matrix containing process data, where the process has n discrete points in each period and the data for m consecutive periods are arranged in the consecutive rows. The objective is to predict the (m+1)th row of X.

The prediction is performed as follows:

(1) Form **A** by appending required number of zeros to each row of X such that the row length is 2^p, p being a positive integer.

(2) Compute WHT of **A**, given by

$$\mathbf{A}_w^T = \frac{1}{n}H_n\mathbf{A}^T, \qquad n = 2^p.$$

(3) Determine the columns of $\mathbf{A}_w$ which are relatively dominant. Model the series of successive elements of each of these columns of $\mathbf{A}_w$ as an autoregressive process; estimate the parameter values, and produce predicted values of the series using the estimated parameters. For the nondominant columns of $\mathbf{A}_w$, the m-th column element may be assumed to be the predicted value of the (m+1)th element. Thus the predicted (m+1)th row of $\mathbf{A}_w$ is produced.

(4) On reverse WHT, the (m+1)th extrapolated row of $\mathbf{A}_w$ will represent the predicted (m+1)th period (where the first n data elements are considered).

7.5.2 Prediction of Power Load on a Substation

The electrical power load on a substation principally shows two periodicities: the daily periodicity and the weekly periodicity. A further periodicity related to the yearly variations is often present. Usually the daily periodicity is most important, which is addressed in this example. The objective is to model the 24-hourly electrical load variations of a substation on Mondays and to produce *one week ahead prediction* of the 24-hourly power demand. Hourly power load data for 10 consecutive Mondays are used for this study, and the load for the 11th Monday is predicted.

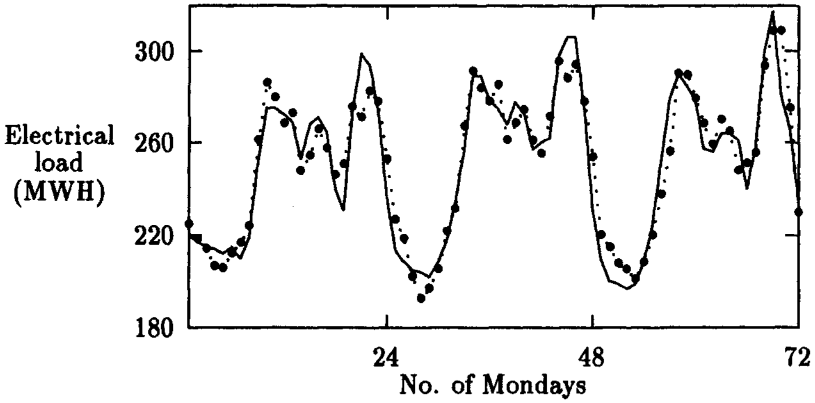


Figure 7.5.1 Reconstruction and prediction of eletrical power load data for 9th to 11th Mondays using WHT; the first two are reconstructed periods, and the last period is the one week ahead prediction.

The available hourly data for the consecutive Mondays are sequentially arranged into the rows of a matrix **A** and each row is extended from 24 to 32 with appended zeros. The 10x32 matrix **A** is WH-transformed to $\mathbf{A}_w$. It is found that 11 columns of $\mathbf{A}_w$ are relatively dominant. The successive elements of each of these columns are treated as separate time series {x(k)}, and are modelled as

$$x(k) = f_0 + f_1 x(k-1) + f_2 x(k-2) + e(k), \qquad (7.5.1)$$

where e(k) is the noise term; the parameters are estimated using the least squares estimator; the parameter values and other details are given in Appendix 7A.1. For each sequence, one-step ahead prediction is obtained as

$$\hat{x}(k+1|k) = f_0 + f_1 x(k) + f_2 x(k-1). \qquad (7.5.2)$$

The predicted 11th row of $\mathbf{A}_w$ is reverse transformed, the first 24 points of which are the predicted values of the power load of the following Monday.

The results are shown in Fig.7.5.1; the data for the 9th and the 10th years are reconstructed using 11 dominant columns of $\mathbf{A}_w$ and the prediction of the 11th year is produced as detailed above. The MSE for prediction works out to be 114.051.

Remarks

(i) WHT can be applied for the periodic prediction of nearly repetitive series. If the periodic pattern or the magnitude over the periods vary, better predictions may be produced using SVD based methods (see Secs.7.8 and 11.2).

(ii) The main advantage of WHT is the computational simplicity and the hardware implementability, although the degree of compaction of information through WHT is not as much as in SVD.

7.6 SINGULAR VALUE DECOMPOSITION (SVD)

Singular value decomposition is an optimal orthogonal decomposition which finds wide applications in rank determination and inversion of matrices, as well as in the modelling, prediction, filtering and information compression of data sequences. SVD is closely related to Karhunen-Loève transformation, singular values being uniquely related to eigenvalues, although the computational requirements of SVD are less than KLT. From a numerical point of view, SVD is extremely robust, and the singular values in SVD can be computed with greater computational accuracy than eigenvalues.

SVD is popularly used for the solution of least squares problems; it offers an unambiguous way of handling rank deficient or nearly rank deficient least squares problems. SVD is also the most definitive method for the detection of the rank of a matrix or the nearness of a matrix to loss of rank.

In this book, SVD has been widely used for modelling and prediction as well as for algebraic matrix operations.

7.6.1 Introduction to Singular Value Decomposition

Given any mxn real matrix **A**, there exist an mxm real orthogonal matrix **U**, an nxn real orthogonal matrix **V** and an mxn diagonal matrix **S**, such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \qquad \mathbf{S} = \mathbf{U}^T\mathbf{A}\mathbf{V}, \tag{7.6.1}$$

where the elements of S can be arranged in nonincreasing order, that is

(i) for a nonsingular $\mathbf{A}$,

$S = \text{diag}\,\{s_1, s_2, \ldots, s_p\}$, $\quad p = \min\,(m,n)$,

$s_1 \geq s_2 \geq s_3 \ldots \geq s_p > 0$, $\quad$ or

(ii) for $\mathbf{A}$ of rank r,

$s_1 \geq s_2 \geq \ldots \geq s_r > 0$ and $s_{r+1} = s_{r+2} = \ldots = s_p = 0$.

In other words, $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$, and

$$S = \begin{bmatrix} s_1 & & 0 \\ & s_2 \ddots & \\ 0 & & s_p \end{bmatrix}, \qquad \text{for } m>n=p.$$

The decomposition (7.6.1) is called the *singular value decomposition.* For proof of this theorem see Golub and Van Loan (1989, p.71). The numbers s_1, s_2, $s_3, \ldots, s_p$ are the *singular values* (or principal values) of $\mathbf{A}$. $\mathbf{U}$ and $\mathbf{V}$ are called the left and right *singular vector* matrices of $\mathbf{A}$ respectively. $\mathbf{U}$ and $\mathbf{V}$ can be expressed as

$\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2 \ldots \mathbf{u}_i \ldots \mathbf{u}_m]$, $\quad$ and

$\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2 \ldots \mathbf{v}_i \ldots \mathbf{v}_n]$,

where for i=1 to p, the m-column vector $\mathbf{u}_i$ and the n-column vector $\mathbf{v}_i$, which correspond to the i-th *singular* value s_i, are called the i-th *left singular vector* and the i-th *right singular vector* respectively.

Again

$$\mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T,$$
$$= \mathbf{U}\mathbf{S}^2\mathbf{U}^T. \qquad (7.6.2)$$

Hence, the columns of $\mathbf{U}$ are the m orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^T$; the diagonal matrix $\mathbf{S}^2$ is the eigenvalue matrix of $\mathbf{A}\mathbf{A}^T$ having the (positive) eigenvalues, s_1^2, $s_2^2, \ldots, s_p^2$, on the diagonal. Similarly

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{S}^2\mathbf{V}^T. \qquad (7.6.3)$$

So the columns of $\mathbf{V}$ are the n orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$, and the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are the same as those of $\mathbf{A}\mathbf{A}^T$. Thus the singular values of $\mathbf{A}$: s_1, $s_2, \ldots, s_p$, are the positive square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^T$ or of $\mathbf{A}^T\mathbf{A}$.

The i-th left and right singular vectors, $\mathbf{u}_i$ and $\mathbf{v}_i$

respectively, corresponding to the i-th singular value s_i are unique.

Remark: Eigenvalue-eigenvector decomposition
With reference to (7.6.2) and (7.6.3), for any symmetric matrix **X**, if $\mathbf{X} = \mathbf{QDQ}^T$, where **Q** is orthogonal and **D** is a real diagonal matrix, then the diagonal elements of **D** are the eigenvalues of **X**, and the column vectors of **Q** are the eigenvectors of **X**. The i-th eigenvector $\mathbf{q}_i$ is associated with the i-th eigenvalue d_i, satisfying the relationship, $\mathbf{Xq}_i = d_i\mathbf{q}_i$. The decomposition $\mathbf{X} = \mathbf{QDQ}^T$ is called the *spectral decomposition* of **X**.

7.6.2 Characteristic Features of SVD

Basic features

(i) The number of nonzero singular values of a matrix **A** is the rank of **A**.

(ii) Both the 2-norm and the Frobenius norm can be defined in terms of the singular values:

$$\|\mathbf{A}\|_2 = s_1, \text{ the largest singular value of } \mathbf{A}; \qquad (7.6.4)$$

$$\|\mathbf{A}\|_F = (s_1^2 + s_2^2 + \ldots + s_p^2)^{1/2}, \qquad p = \min\,(m,n). \qquad (7.6.5a)$$

(iii) From energy considerations,

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2 = \|\mathbf{A}\|_F^2 = s_1^2 + s_2^2 + \ldots + s_p^2, \qquad (7.6.5b)$$

a_{ij} being the jth element of the i-th row of **A**, and since

$$\mathbf{A} = \mathbf{USV}^T = \sum_{i=1}^{p} \mathbf{u}_i s_i \mathbf{v}_i^T,$$

($\mathbf{u}_i$ and $\mathbf{v}_i$ being the columns of **U** and **V** respectively), the energy in the decomposed component matrix $\mathbf{u}_i s_i \mathbf{v}_i^T = s_i^2$.

Rank characterization

The number of nonzero singular values indicate the effective rank of a matrix. The singular values also indicate precisely how close a given matrix is to a matrix of lower rank which can be explained as follows.

For an mxn full-rank matrix **A**,

$$\mathbf{U}^T\mathbf{AV} = \text{diag}\,(s_1, s_2, \ldots, s_p), \qquad p = \min\,(m,n).$$

Let $\mathbf{A}_r$ be a close lower (say r) rank matrix: r<p. So

$$U^T\mathbf{A}_r\mathbf{V} = \text{diag}\ (s_1,\ s_2,\dots,\ s_r,\ 0,\dots,\ 0).$$

Hence

$$U^T[\mathbf{A}-\mathbf{A}_r]\mathbf{V} = \text{diag}\ (0,\dots,0,\ s_{r+1},\dots,\ s_p).$$

Since the matrix 2-norm is equal to the highest singular value, following (7.6.4),

$$\|\mathbf{A}-\mathbf{A}_r\|_2 = s_{r+1}.$$

Hence the smallest singular value of **A** is the 2-norm distance of **A** to the set of all rank deficient matrices.

Effects of changes of matrix size

The ratio of the largest singular value of **A** to the smallest nonzero singular value of **A** is called the *condition number.*

If a column is added to the mxn matrix **A** where $m \geq n$ the largest singular value increases and the smallest singular value decreases, and hence the condition number increases; the result of deletion of a column is the opposite. If m<n, with addition of a column the largest singular value increases but the smallest singular value does not decrease.

Stability against perturbations

Singular values of a matrix show high degree of stability. Perturbations in the elements of a matrix can cause perturbations of equal or smaller magnitudes in the singular values as follows.

If the mxn matrix **A** $(m \geq n)$ is perturbed by $\tilde{\mathbf{A}}$ resulting in $\mathbf{A}^*$,

$$\mathbf{A}^* = \mathbf{A} + \tilde{\mathbf{A}}, \qquad \text{then}$$

$$|s_i^*-s_i| \leq \tilde{s}_1 = \|\tilde{\mathbf{A}}\|,$$

where s_i^* , s_i and $\tilde{s}_i$ are the singular values of $\mathbf{A}^*$, **A** and $\tilde{\mathbf{A}}$ respectively, and $\tilde{s}_1$ is the largest singular value of $\tilde{\mathbf{A}}$. The limit on the perturbation in the singular values can also be expressed by the Wielandt-Hoffman theorem:

$$\sum_{i=1}^{n}(s_i^*-s_i)^2 \leq \sum_{i=1}^{n}\tilde{s}_i^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{a}_{ij} = \|\tilde{\mathbf{A}}\|_F^2,$$

$\tilde{a}_{ij}$ being the jth element of the i-th row of $\tilde{\mathbf{A}}$.

Numerical robustness and implementation

Singular value decomposition is extremely robust numerically compared with eigenvalue computations. Computer programs for SVD are available in many high quality software packages, e.g., LINPACK, EISPACK and MATLAB. The coding in FORTRAN is also given in Lawson and Hanson (1974).

Example 7.6.2 Perform the singular value decomposition of the matrix **A** given by

$$\mathbf{A} = \begin{bmatrix} 3 & 9 \\ 4 & 2 \\ 7 & 3 \end{bmatrix}.$$

SVD of **A** produces

$$\mathbf{U} = \begin{bmatrix} 0.7436 & 0.6678 & 0.0324 \\ 0.3455 & -0.3424 & -0.8737 \\ 0.5724 & -0.6609 & 0.4854 \end{bmatrix}, \quad \mathbf{V}^T = \begin{bmatrix} 0.6420 & -0.7667 \\ 0.7667 & 0.6420 \end{bmatrix},$$

and the singular values: $s_1 = 11.8695$, $s_2 = 5.2071$. Note that conforming to (7.6.5), the energy in **A** is 168.

Remark: One important feature of SVD is the ability to characterize the periodicity present in the data, which is discussed next. The use of SVD for determination of the periodicity in the data is treated in Sec.11.4.

7.7 CHARACTERIZATION OF PERIODIC PROCESSES USING SVD

SVD offers one of the most robust approaches to the analysis, modelling and prediction of data series with periodic excursions. In this section, the characterization of the periodic series in terms of the decomposition components **U**, **S**, and **V** is considered.

Arranging the data

For SVD based analysis, the data have to be arranged into a matrix. Consider a process or data sequence:

$$\{x(.)\} = \{x(1), \quad x(2),\ldots\}.$$

If the series {x(.)} is periodic, with period length n, M consecutive n-long periods can be arranged into a matrix **X**, such that the consecutive periods occupy the consecutive rows of X, as follows:

$$\mathbf{X} = \begin{bmatrix} x(1) & x(2) & \dots & x(n) \\ x(n+1) & x(n+2) & \dots & x(2n) \\ \vdots & \vdots & \ddots & \vdots \\ x((M-1)n+1) & x((M-1)n+2) & \dots & x(Mn) \end{bmatrix}. \qquad (7.7.1)$$

SVD of the Mxn matrix X is given by

$$\mathbf{X} = \mathbf{USV}^T, \qquad (7.7.2)$$

where $\mathbf{U} = [\mathbf{u}_1,\dots,\mathbf{u}_M]$, and $\mathbf{V} = [\mathbf{v}_1,\dots,\mathbf{v}_n]$ are MxM and nxn orthogonal matrices respectively, and S is a diagonal matrix of singular values s_1 to s_p, with $s_1 \geq \dots \geq s_p \geq 0$, $p = \min(M,n)$.

The left ($\mathbf{u}_1$ etc.) and the right ($\mathbf{v}_1$ etc.) singular vectors form a basis for the column-space and the row-space of X respectively.

Characterization

Consider the following types of periodic processes.

Case 1 The series {x(.)} is *strictly periodic* with period length n, that is

$$x(k) = x(k+n).$$

Here, all the rows of X will be identical, and X will be a rank-one matrix. SVD of X will produce *one* nonzero singular value s_1, all other singular values being zero. Hence $\mathbf{X} = \mathbf{u}_1 s_1 \mathbf{v}_1^T$. $\mathbf{v}_1^T$, the first row of $\mathbf{V}^T$, will represent the pattern or the normalized distribution of the series over one period; The elements of the M-vector $\mathbf{u}_1 s_1$ represent the scaling factors for each row of the data matrix X.

Let

$$\mathbf{u}_1 s_1 = \mathbf{z}_1 = [z_{11}\ z_{21}\ \dots\ z_{i1}\ \dots z_{M1}]^T. \qquad (7.7.3)$$

Thus the i-th row of X will be given by $z_{i1}\mathbf{v}_1^T$. Since {x(.)} is perfectly periodic, the M elements of $\mathbf{z}_1$ will be identical. See Example 7.7(1).

Case 2 The series {x(.)} is *nearly periodic* with fixed period length n but x(k) is not necessarily equal to x(k+n).

There are two main possibilities:

(a) The series {x(.)} has the same repeating pattern but

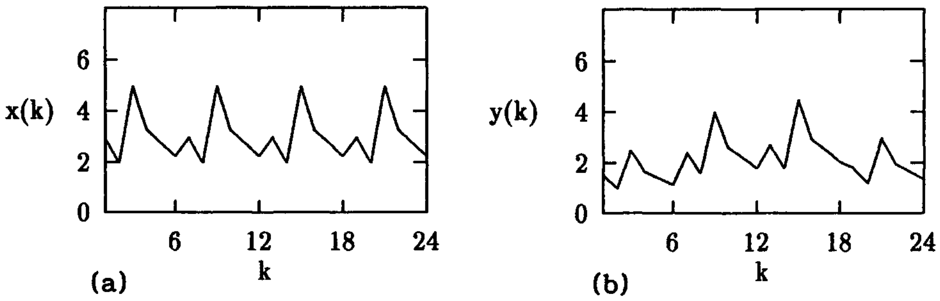


Figure 7.7.1 (a) A strictly periodic series, and (b) a nearly periodic series with repeating pattern.

with different amplitudes over different periods.

In such a case the rows of **X** will be linear functions of each other, and hence **X** will still be a rank-one matrix. $\mathbf{v}_1^T$ will represent the periodic pattern but the elements of $\mathbf{z}_1$ in (7.7.3) will be different. See Example 7.7(1).

(b) The series {x(.)} has a nearly repeating pattern with different amplitudes over different periods.

In such a case, **X** can be a full rank matrix. SVD of **X** will still produce one dominant singular value s_1, the other singular values being insignificantly small: $s_1 \gg s_2$. So the rank-one approximation, $\mathbf{X} \simeq \mathbf{u}_1 s_1 \mathbf{v}_1^T$ will be permissible. The Airline traffic series discussed in Example 7.7(2) belongs to this category.

Example 7.7(1) SVD characterization of two repeating periodic series

Fig.7.7.1(a) shows a strictly periodic series {x(.)}, and Fig.7.7.1(b) shows a periodic series {y(.)} with a repeating pattern, which is differently scaled over different periods. The 4 consecutive periods of {x(.)} and {y(.)} are arranged into matrices **A** and **B** respectively, and are singular value decomposed as follows.

$$\mathbf{A} = \begin{bmatrix} 3.00 & 2.00 & 5.00 & 3.25 & 2.75 & 2.25 \\ 3.00 & 2.00 & 5.00 & 3.25 & 2.75 & 2.25 \\ 3.00 & 2.00 & 5.00 & 3.25 & 2.75 & 2.25 \\ 3.00 & 2.00 & 5.00 & 3.25 & 2.75 & 2.25 \end{bmatrix}$$

$\mathbf{A} = \mathbf{U}_A \mathbf{S}_A \mathbf{V}_A^T$, where

$\mathbf{S}_A = \text{diag}\ \{15.6445 \quad 0 \quad 0 \quad 0{:}0\}$,

$$\mathbf{U_A} = \begin{bmatrix} 0.5000 & -0.5000 & -0.5000 & -0.5000 \\ 0.5000 & 0.8333 & -0.1667 & -0.1667 \\ 0.5000 & -0.1667 & 0.8333 & -0.1667 \\ 0.5000 & -0.1667 & -0.1667 & 0.8333 \end{bmatrix},$$

$$\mathbf{V_A} = \begin{bmatrix} 0.3835 & -0.9235 & 0 & 0 & 0 & 0 \\ 0.2557 & 0.1062 & -0.6921 & -0.4499 & -0.3807 & -0.3115 \\ 0.6392 & 0.2654 & 0.6248 & -0.2439 & -0.2063 & -0.1688 \\ 0.4155 & 0.1725 & -0.2439 & 0.8415 & -0.1341 & -0.1097 \\ 0.3516 & 0.1460 & -0.2063 & -0.1341 & 0.8865 & -0.0929 \\ 0.2876 & 0.1195 & -0.1688 & -0.1097 & -0.0929 & 0.9240 \end{bmatrix}.$$

Again

$$\mathbf{B} = \begin{bmatrix} 1.50 & 1.00 & 2.50 & 1.625 & 1.375 & 1.125 \\ 2.40 & 1.60 & 4.00 & 2.600 & 2.200 & 1.800 \\ 2.70 & 1.80 & 4.50 & 2.925 & 2.475 & 2.025 \\ 1.80 & 1.20 & 3.00 & 1.950 & 1.650 & 1.350 \end{bmatrix}.$$

$\mathbf{B} = \mathbf{U_B}\mathbf{S_B}\mathbf{V_B}^T$, where

$\mathbf{S_B} = \text{diag }\{11.2270 \quad 0 \quad 0 \quad 0{:}0\}$,

$$\mathbf{U_B} = \begin{bmatrix} 0.3484 & -0.9326 & -0.0837 & -0.0440 \\ 0.5574 & 0.2807 & -0.7795 & -0.0534 \\ 0.6271 & 0.1640 & 0.4661 & 0.6022 \\ 0.4180 & 0.1569 & 0.4099 & -0.7954 \end{bmatrix},$$

$$\mathbf{V_B} = \begin{bmatrix} 0.3835 & -0.6286 & 0.6693 & -0.0236 & 0.0963 & 0 \\ 0.2557 & 0.1344 & -0.1350 & 0.3351 & 0.8791 & -0.1149 \\ 0.6392 & 0.5996 & 0.2495 & 0.1614 & -0.3260 & -0.1930 \\ 0.4155 & -0.2935 & -0.5226 & -0.4550 & -0.0493 & -0.5086 \\ 0.3516 & -0.3185 & -0.4445 & 0.5277 & -0.2601 & 0.4809 \\ 0.2876 & 0.1993 & -0.0286 & -0.6129 & 0.2037 & 0.6780 \end{bmatrix}.$$

Note that **A** and **B** are rank-one matrices, which also shows from both $\mathbf{S_A}$ and $\mathbf{S_B}$ having one nonzero singular value. Since $\{x(.)\}$ is strictly periodic, all elements of the first column of $\mathbf{U_A}$ are the same. The first row of $\mathbf{V_B^T}$ gives the pattern of $\{y(.)\}$, which is scaled by the elements of the vector $\mathbf{u_{B1}}s_{B1}$, where $\mathbf{u_{B1}}$ is the first column of $\mathbf{U_B}$, and s_{B1} is the first singular value of $\mathbf{S_B}$ (=11.2270).

Summary

If a periodic series $\{x(.)\}$ is arranged into a matrix X, with the periods aligned into the rows of X, and if $X = USV^T$,
(i) v_1^T represents the periodic pattern of $\{x(.)\}$, and the elements of $u_1s_1 = z_1 = [z_{11}\ z_{21}\ \dots\ z_{i1}\ \dots z_{M1}]^T$, will be the scaling factors.
(ii) if $\{x(.)\}$ has a strictly repeating pattern, only s_1 is nonzero while $s_2 = \dots = s_p = 0$; the i-th row of X will be given by $z_{i1}v_1^T$.
(iii) if $\{x(.)\}$ is periodic with a nearly repeating pattern, $s_1 >> s_2$; the i-th row of X will be given by $z_{i1}v_1^T$ through rank one approximation of X, that is $X = USV^T \simeq u_1s_1v_1^T$.

Degree of periodicity

The total energy in X is given by

$$Q_X = \sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2 = \|A\|_F^2 = s_1^2 + s_2^2 + \dots + s_p^2.$$

The degree of periodicity depends on the percentage of the total energy contained in the most dominant decomposition component (that is s_1^2). So $\mathcal{P}$, the degree of periodicity, may be given by

$$\mathcal{P} = s_1^2/s_2^2. \tag{7.7.4}$$

An alternative expression for periodicity can be

$$\mathcal{P}' = s_1^2 \Big/ \sum_{i=1}^{p} s_i^2. \tag{7.7.5}$$

For a perfectly periodic series, $\mathcal{P}$ is ∞ while $\mathcal{P}' = 1$. The closeness of a nearly periodic series to the rank-one approximated series can be assessed using (7.7.4) and (7.7.5). Assessment of the degree of periodicity is central to the concept of periodic decomposition discussed in Sec.11.5 and the SVR spectrum discussed in Appendix 11.

Remark
The term 'nearly periodic', in the present context, refers to deviation from periodicity in terms of either the pattern, the period length, or the magnitudes over the periods. The SVD based characterization of nearness to periodicity is an assessment in terms of rank-oneness of the appropriately configured data matrix. In mathematics, a closely related term is 'almost periodic functions', which

concerns the period lengths being almost the same; such functions have been rigorously treated in Corduneanu (1968).

Example 7.7(2) SVD analysis of nearly periodic series with fixed periodicity

Consider SVD analysis of
(i) the Trans-Atlantic Airline Passenger series, and
(ii) the Homogeneous Indian Rainfall series.
Both the series contain monthly data and are nearly periodic with (fixed) yearly periodicity.

For the Trans-Atlantic Airline Passenger series shown in Fig.4.3.3, a 4x12 moving window, say **A**(k), is considered moving over a 12x12 data set **X**, containing monthly data for 1949 to 1960 (see Appendix 7A.2); so **A**(1) will be composed of the first 4 rows of **X**, **A**(2) will be composed of 2nd to 5th rows of **X** and so on. The singular values are computed for each **A**(k), for k = 1 to 6. Both the data and the singular values are given in Appendix 7B.2. The progressive distribution of the singular values of {**A**(k)}, normalized by s_1 for each window k, is shown in Fig.7.7.2.

A similar exercise is performed on the homogeneous Indian Rainfall series (Appendix 7F) shown in Fig.2.2.1. The data from 1940 to 1959 are used here; a 10x12 **A**(k) is considered in this case. The distribution of the singular values (normalized by s_1) obtained from the SVD of **A**(k), for k = 1 to 6, is shown in Fig.7.7.3.

For the Airline Passenger series, a sharp edge in Fig. 7.7.2 indicates the presence of one dominant singular value for the different windows {**A**(k)} over the data series, whereas the distribution of the singular values is not that sharp for the rainfall series, which indicates that the degree of repetitiveness of the periods in the rainfall series to be comparatively low.

Similar analysis applied to the quasiperiodic processes appears in Sec.11.2.

7.8 MODELLING AND PREDICTION USING SVD

It is assumed that the periodic data {x(.)} are arranged into an Mxn matrix **X** as discussed in the last section. The objective is to model the series {x(.)} and to produce one-period ahead prediction, that is predict the (M+1)th row of **X**.

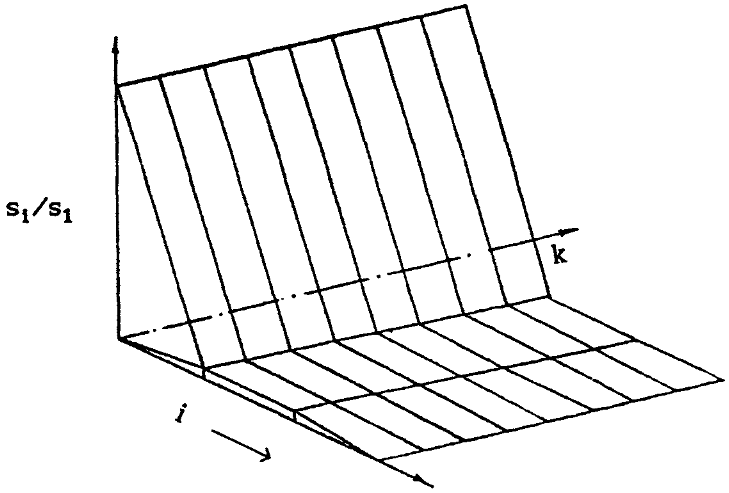


Figure 7.7.2 The progressive distribution of normalized singular values for the Airline traffic series.

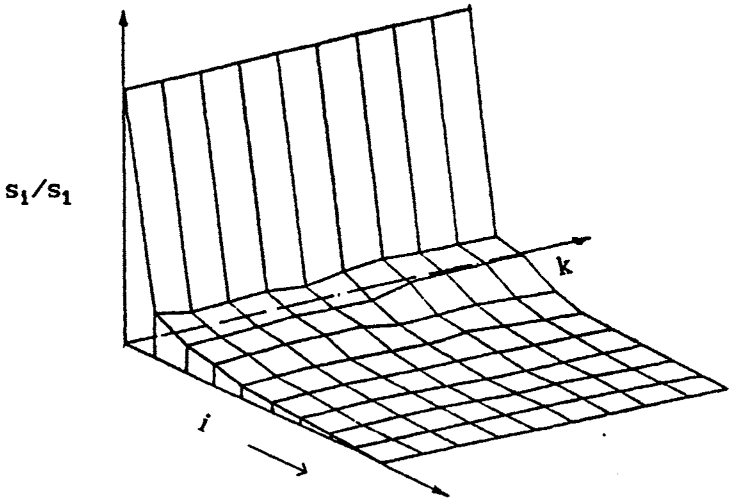


Figure 7.7.3 The progressive distribution of normalized singular values for the Rainfall series.

7.8.1 Principle of Modelling

Consider an mxn data window moving (downwards) over X (where m<M); let the so formed matrix sequence be {A(i)}. The k-th data window forms A(k), which is composed of the last m rows of X. For each value of k, A(k) is singular value decomposed, and the series {x(.)} is modelled in terms of the so obtained SVD components.

Let the SVD of A(i) be given by

$$A(i) = U(i)S(i)V^T(i).$$

Periodic prediction can be computed as follows. Let A(1),...,A(k) be the consecutive mxn data windows of X. The singular value decomposition of A(k) is performed for each value of k.

The modelling and prediction policy is based on two main assumptions:

(i) only the first singular value of A(k) is predominant, for any k, and

(ii) $v_1^T(k)$ can be assumed to remain almost unchanged between the two consecutive positions A(k-1) and A(k).

The case, when more than one singular value is dominant, is discussed in the Remarks later in this section.

Define

$$z_{m1}(k) = u_{m1}(k)s_1(k). \tag{7.8.1}$$

The propagation of the sequence $\{z_{m1}(k)\}$ can be represented by any suitable discrete-time model. Typically an AR model may be considered:

$$F(q^{-1})z_{m1}(k) = e(k), \tag{7.8.2}$$

where

$$F(q^{-1}) = 1 + f_1q^{-1} + f_2q^{-2} + \dots + f_Nq^{-N},$$

and {e(k)} is a sequence of uncorrelated noise. The model (7.8.2) may be used to produce the prediction $\hat{z}_{m1}(k+1|k)$; the consequent the prediction of one complete period following A(k) is given by

$$\hat{a}_{(m+1)1}(k+1|k) = \hat{z}_{m1}(k+1|k)v_1^T(k). \tag{7.8.3}$$

Summary

(1) Form Mxn matrix X from M consecutive periods each of length n.

(2) Choose a moving data window size mxn for A(k).

(3) Starting from the top of X move the data window A(k)

over X, one row down at a time; for each k, perform SVD of A(k) and store $z_{m1}(k) = u_{m1}(k)s_1(i)$.

(4) Model $\{z_{m1}(k)\}$ process (7.8.2) and produce one-step prediction $\hat{z}_{m1}(k+1|k)$.

(5) Compute one-period prediction: $\hat{z}_{m1}(k+1|k)v_1(k)$.

Remarks

(1) The assumption that the first singular value is dominant is not a limitation. If the second singular value is also dominant, and if $v_2(k)$ remains reasonably unchanged between two successive windows k and k+1, the same procedure may be applied to predict $\hat{z}_{m2}(k+1|k)v_2(k)$; the overall prediction will be $\hat{z}_{m1}(k+1|k)v_1(k) + \hat{z}_{m2}(k+1|k)v_2(k)$. Thus any number of additive components may be accommodated in the prediction.

(2) The choice of the data window length m is restricted by the amount and the nature of data available; the highest value can be typically <(M−6); the smallest value should be less than or equal to the number of periods over which the dynamics of the process remains relatively steady.

7.8.2 Case Study: Periodic Prediction of Airline Traffic

The Airline traffic series has been discussed in Example 7.7(2) in Sec.7.7. The objective is to produce periodic (i.e. yearly or one- to 12-step ahead) prediction of this monthly data series and to study the robustness of the prediction procedure.

Prediction

Here n = 12, and m is chosen to be 4. The data for the first 10 years are used to produce the prediction for 11th year, and the data up to the 11th year are used to predict the 12th year. The data are (natural-)logarithmically transformed before performing SVD to reduce the variance and periodic variability in the data.

The singular values for A(1) to A(7) show that s_1 is dominant. The overlaid plot of $v_1(1)$ to $v_1(7)$ in Fig.7.8.1 shows $v_1(k)$ remaining reasonably steady over the different data windows. So the two basic assumptions for the present method of modelling as stated in Sec.7.8.1 are valid.

The first 10 periods yield only 7 data points for $\{z_{m1}(k)\}$ sequence which is modelled as

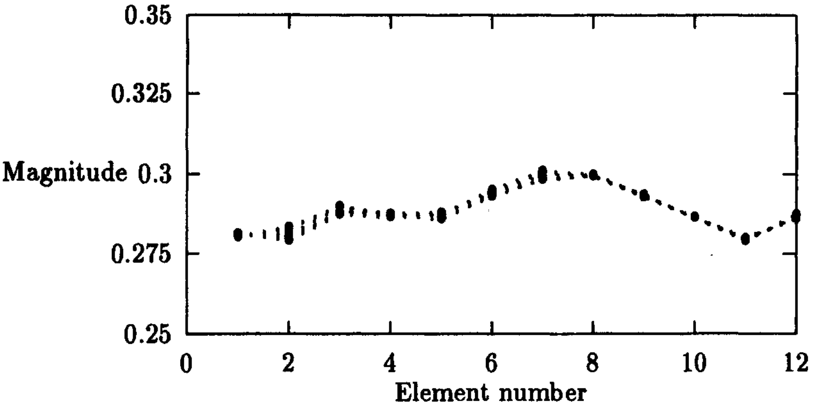


Figure 7.8.1 Overlaid plot of $v_1(1)$ to $v_1(7)$ for log-transformed Airline traffic series.

$$z_{m1}(k) = f_0 + f_1 z_{m1}(k-1) + e(k), \qquad m = 4, \qquad (7.8.4)$$

where the least squares estimates for f_0 and f_1 work out as $f_0 = 2.197$ and $f_1 = 0.965$. $\hat{z}_{m1}(11|10)v_1(10)$, the so produced prediction is antilog transformed to produce $\hat{a}_{m1}(11|10)$. The procedure is repeated to produce $\hat{a}_{m1}(12|11)$. The prediction results are shown in Fig.7.8.2; the MSE per sample is found to be 354.28 for the predicted series.

Remarks: Since the present method is based on the decomposition components corresponding to the most dominant singular value, it can be used even if the data are contaminated with noise; this is because aperiodic noise will be largely associated with the smaller singular values.

Robustness

The robustness of the modelling and prediction procedure against additive white Gaussian noise is now considered; the performance of the SVD based method is compared with that of a multiplicative model.

The log-transformed Airline traffic series has been modelled by Box and Jenkins (BJ) (1976, p.306) as

$$z(k) = (1-\theta_1 q^{-1})(1-\theta_2 q^{-2})e(k), \qquad (7.8.5)$$

where the estimated parameters of this multiplicative model are stated to be $\theta_1 = 0.4$ and $\theta_2 = 0.6$.

The noise is added to the untransformed data. The parameters of (7.8.5) are estimated with 20 different levels

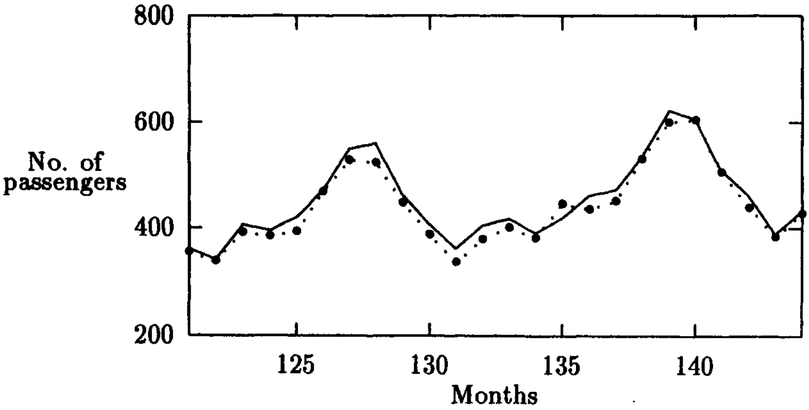


Figure 7.8.2 One- to 24-step ahead prediction of the Airline traffic series for the 11th and the 12th years.

of noise by finding the minimum of the error sum of squares surface. For each noise level 1000 different noise patterns are tried for each of which the parameters are estimated; the averaged parameter values are used for computing prediction. The MSE for one- to 12-step ahead prediction by both the methods are computed for the 11th and the 12th years. As shown in Fig.7.8.3, while at high signal to noise ratio (SNR), the performances of the SVD based method and

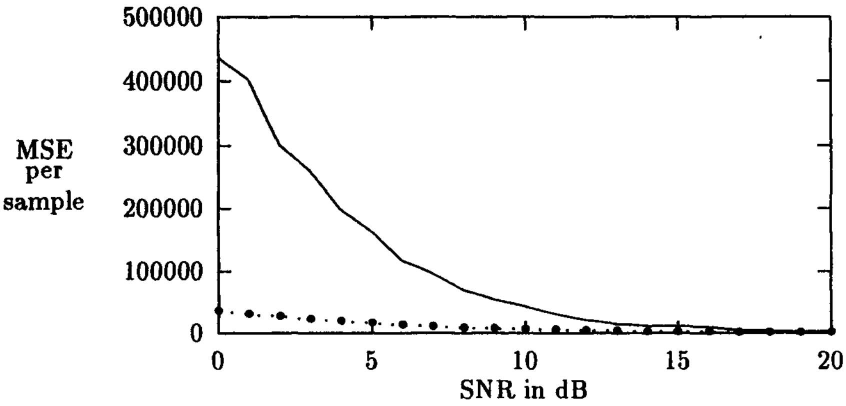


Figure 7.8.3 MSE vs. signal to noise ratio for periodic prediction using SVD based and BJ method (————) on Air traffic series with added white Gaussian noise.

the BJ method are comparable, at low SNRs the SVD based method performs much better; the reason is that the effect of noise is largely confined to the smaller singular values which are ignored in the SVD based method.

7.9 CONCLUSIONS

The property of compaction of information through orthogonal transformation has been used for modelling and prediction of nearly periodic series.

In this chapter, first the subject of orthogonal transformation was introduced and the optimal transformation through KLT was presented. Next the Walsh-Hadamard transform (WHT) was discussed. The rest and the major part of this chapter has been devoted to the study of singular value decomposition (SVD). Both WHT and SVD offer efficient procedures for the modelling and prediction of nearly periodic processes. Particularly the SVD based approach is very powerful and numerically stable.

The main attraction of WHT is that it is computationally simple and is even hardware implementable. WHT operates on vectors rather than matrices which is also an advantage. The transformed components are modelled, rather than the time domain data. However, the degree of compaction of information being much less compared with SVD, more data are required for sensible modelling.

SVD has been widely used in the present text both for modelling and analysis (in Chapters 3, 7, 8, 9, 10, 11 and 14) as well as a tool for matrix operations (in Chapters 3, and 12). Following an introduction to SVD and its properties, the analysis and modelling of nearly periodic series using SVD has been detailed in this chapter.

From numerical point of view, the two attractive features of SVD are (i) that it can be computed in a numerically stable way, and (ii) that the singular values are well conditioned. SVD offers the most robust and definitive method for the determination of the rank of a matrix. For the modelling of periodic signals, the special structural feature of SVD can be made use of. The data are configured into a matrix with the consecutive periods occupying the consecutive rows of the matrix. The near rank-oneness of the matrix reveals the periodic nature of the data series. The series is modelled through the linear modelling of the decomposition components, which contain maximally compressed information about the periodic data.

Besides the numerical robustness, the proposed method of modelling and prediction is applicable even if the data are noisy and the period length as well as the periodic pattern varies to certain extent. The present discussions have been confined to processes with fixed period length only; processes with varying period length are studied in Chapter 11.

It should be possible to automate the modelling and prediction procedures using both SVD as well as WHT, as operator expertise is not a prerequisite.

REFERENCES

Remark: Orthogonal transformation and KLT are studied in detail in [1]. WHT was introduced in [24]. A standard text on WHT is [3]; the theory and applications of WHT are also covered in [1,2,19,23]. Hadamard matrices are discussed in [17]. Prediction applications of WHT appear in [4,18]. Authoritative references on SVD and orthogonal transformation are [9,14,25]; the coverage of [9] is most wide. The robustness and computational aspects of SVD are also discussed in [12,13]; rank assessment is addressed in [21]. A lucid introductory description of SVD features in the book on algebra [22]. Various applications of SVD also feature in [7]. Prediction applications of SVD appear in [10,16]. The use of SVD features widely in this book, for example in Chapters 3,10,11,12,13 and 15.

LINPACK [8] is a collection of machine-independent, efficient FORTRAN programs for linear algebra, which is available free of cost. EISPACK [20] is a similar package for eigenvalue problems. MATLAB [15] is a powerful package for interactive matrix computations, where graphics facilities are also included.

[1] Ahmed, N., and K.R. Rao (1975): *Orthogonal Transformation for Digital Signal Processing*, Springer-Verlag, Berlin.

[2] Alexandridis, N.A. (1989): *Image Processing in Supercomputing Systems, in Designing and Programming Modern Computer Systems* Vol. III Parallel Architectures, Networks, Applications, Eds. S.P. Kartashev and S.I. Kartashev, Prentice-Hall, Eaglewood Chiffs, 323-400.

[3] Beauchamp, K.G. (1984): *Applications of Walsh and Related Functions with an Introduction to Sequency Theory*, Academic Press, London.

[4] Bhattacharya, T.K., and T.K.Basu (1991): 'Medium range forecasting of hourly power system load by time series analysis using Walsh transform', *Electrical Power & Energy Systems*, 13(4), 193–200.

[5] Box, G.E.P., and G.M. Jenkins (1976): *Time Series Analysis: Forcasting and Control*, Holden-day, San Francisco, CA.

[6] Corduneanu, C. (1968): *Almost Periodic Functions*, Wiley, New York.

[7] Deprettere, F. (Ed.) (1988): *SVD and Signal Processing: Algorithms, Applications and Architectures*, North-Holland, Amsterdam.

[8] Dongarra, J.J., C.B. Moler, J.R. Bunch, and G.W. Stewart (1979): *LINPACK User's Guide*, SIAM Publications, Philadelphia.

[9] Golub, G.H., and C.F. Van Loan (1989): *Matrix Computations*, 2nd edn., The Johns Hopkins University Press, Baltimore.

[10] Kanjilal, P.P., and S. Palit (1994): 'The singular value decomposition - applied in the modelling and prediction of quasiperiodic processes', *Signal Processing*, 35(3), 257–267.

[11] Keeling, C.D., R.B. Bacastow, and T.P. Whorf (1982): 'Measurement of the concentration of carbon dioxide at Mauna Loa observatory, Hawaii', in *Carbon Dioxide Review: 1982*, Ed. W.C. Clark, Oxford Univ. Press, Oxford, 377–385.

[12] Klema, V.C., and A.J. Laub (1980): 'The singular value decomposition: its computation and some applications', *IEEE Trans. Automatic Control*, AC-25, 164–176.

[13] Laub, A.J. (1985): 'Numerical linear algebra aspects of control design computations', *IEEE Trans. Automatic Control*, AC-30, 727–764.

[14] Lawson C.L., and R.J. Hanson (1974): *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ.

[15] MATLAB matrix software, The MathWorks, Inc., 158 Woodland St, Sherborn, MA 01770.

[16] Palit, S., and P.P. Kanjilal (1994): 'On the singular value decomposition, applied in the analysis and prediction of almost periodic signals', *Signal Processing*, 40(3), 269–285.

[17] Raghavarao, D. (1971): *Construction and Combinational Problems in Design of Experiments*, John Wiley, New York.

[18] Rao, G.P., J.K. Singh, K.A. Rao, and S. Sinha (1993): 'Forecasting for agricultural management', in *Concise Encyclopedia of Environmental Systems*, P.C. Young (Ed.), Pergamon Press, Oxford, 236-242.

[19] Rao, G.P. (1983): *Piecewise Constant Orthogonal Functions and Their Applications to Systems and Control*, Lecture Notes in Control Information Sciences, Vol. 55, Springer Verlag, Berlin.

[20] Smith, B.T., J.M. Boyle, J.J. Dongara, B.S. Garbow, Y. Ikebe, V.C. Klema, and C.B. Moler (1976): *Matrix Eigensystem Routines*, EISPACK Guide, 2nd edn., Springer Verlag, New York.

[21] Stewart, G.W. (1984): 'Rank degeneracy', *Siam J. Sci. and Stat. Comp.*, 5, 403-413.

[22] Strang, G. (1988): *Linear Algebra and its Applications*, 3rd edn., Harcourt Brace Jovanovich, San Diago.

[23] Tzafestas, S.G. (Ed.) (1985): *Walsh Functions in Signals and Systems, Analysis and Design*, Van Nostrand Reinhold Co., New York.

[24] Walsh, J.L. (1923): 'A closed set of orthogonal functions', *American Journal of Mathematics*, 45, 5-24.

[25] Wilkinson, J.H. (1965): *The Algebric Eigenvalue Problem*, Oxford University Press, Oxford.

CHAPTER 8

MODELLING OF NONLINEAR PROCESSES: AN INTRODUCTION

Certain special features characterize a nonlinear process which can be represented by a single-stage or a multistage model, linear or nonlinear in the parameters.

8.1 INTRODUCTION

Most real-life processes are nonlinear to varying extents. A nonlinear process may be simply a time series, for example, the yearly averaged Sunspot activity process in astronomy (Fig.8.2.4) or the river flow discharge in hydrology; or, it may be a complex process with many independent inputs, whose influences on the output are imprecisely known, for example, the economic inflation process, or the quality of molten iron while being tapped from the blast furnace. Since nonlinearity can be of various types and time varying, and also since the data may not be sufficient or adequately informative, both structure selection and parameter estimation for a real-life nonlinear process can be quite complex.

There are many approaches to nonlinear system modelling (Billings, 1980, Haber and Unbehauen, 1990), each trying to capture the nonlinear characteristics through one or more nonlinear functional blocks; modelling usually involves local or piece-wise linearization, otherwise nonlinear optimization is performed. The suitability of any method of modelling and identification largely depends on the nature of the problem and the amount of information available on the process. The cross validation tests (Sec.3.6.4) and the quality of prediction obtained through the model will demonstrate how closely the model represents the underlying process. The present chapter is devoted to the study of (a) the characteristic features of processes with nonlinearity, and (b) the problems concerning modelling of such processes; a summary of some selected classes of nonlinear models is also presented.

The organization of this chapter is as follows. Some basic concepts related to nonlinear processes and their modelling are discussed in Sec.8.2. The special features of processes with nonlinear periodicity are studied in Sec.8.3; a comparative analysis using the state-space diagrams, the singular value decomposition (SVD) based characterization and frequency domain decomposition is also presented. Finally a summary of some selected classes of well-studied models is given in Sec.8.4.

8.2 BASICS OF NONLINEAR PROCESSES

8.2.1 Characteristic Features

Features particular to nonlinear processes are as follows.

(a) Algebraic manifestation:
A basic manifestation of nonlinearity is in the principle of superposition not being valid.

(b) Linearity/nonlinearity in the parameters
A nonlinear process may be represented by a linear-in-the-parameter model:

$$f(x,a) = a_o + a_1x + a_2x^2 + \dots + a_mx^m, \qquad (8.2.1)$$

whereas a nonlinear model is nonlinear in the parameters; for example,

$$f(x,\theta) = \theta_0 + \theta_1x + \theta_2e^{x\theta_3}. \qquad (8.2.2)$$

Nonlinearity of a nonlinear model is manifested in the dependence of $\partial f/\partial\theta_i$ on any of the parameters, θ_i. Thus whether a model is linear or nonlinear depends on how parameters enter the model, rather than how the variables enter the model.

(c) Stationarity and initial condition
Visibly, stationarity implies no growth or decline in the data with time. A stationary time series shows almost constant mean and variance. The autocorrelation function of a stationary series quickly drops close to zero from the maximum, typically after the second or third time lag, whereas the decrease is slow for a nonstationary process.

For a linear process, stationarity is a *global* property, i.e. the process is either stationary or non-stationary over all the space and time. On the other hand, a nonlinear process is at the most *locally* stationary. Besides, unlike the linear process, stability of a nonlinear process is dependent on initial conditions.

(d) Irreversibility

Irreversibility of time series data refers to the probabilistic features not being retained upon reversal of the direction of time. Nonlinear processes are distinctly time-irreversible. A corollary to time-irreversibility is the fact that for a nonlinear system, the output to a normal input is not normal, whereas for a linear system the output is also normal.

(e) Invertibility

If y(t) is the output of a process driven by an unobservable noise process e(t) then the model relating y(t) with e(t) is invertible, if the estimates $\hat{e}(t)$ of e(t) can be produced from y(t) such that the error, $\varepsilon(t) = \hat{e}(t)-e(t)$, tends to zero in some statistical sense as the number of observations y(t) tends to infinity; thus invertibility implies reconstructibility of the input from the output. Nonlinear processes can only be conditionally invertible.

(f) Periodicity

If a time series tends to repeat at a certain time interval, it is referred to as a periodic signal or a periodic process. The three basic attributes of periodicity are the period-length, the pattern over a period and the magnitude of the pattern; if these features remain unchanged, the process is called perfectly periodic, otherwise the process will be aperiodic which can be quasiperiodic or chaotic. A periodic process can be expressed by a linear model, whereas aperiodicity is akin to nonlinearity. Further discussions on nonlinear periodicity follow in Sec.8.3.

(g) Steady-state response

Response of a stable system has two components: a transient and a steady-state component, the latter referring to asymptotic response (i.e. response as time tends to infinity). The transient component is expected to subside with time, leaving only the steady-state component. If the system is linear, the steady-state component will be either constant or periodic. If the system is nonlinear, the steady-state response will be bounded but can be periodic quasiperiodic or chaotic in nature.

8.2.2 Basic Models

Some basic representations of nonlinear processes are discussed here.

Polynomial models

A large class of nonlinear processes can be expressed by polynomial or curvilinear models.

For processes with one independent variable x, a typical model can be:

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m + \xi, \qquad (8.2.3)$$

where ξ is the unmeasurable noise component, and m is the order of the model. For example, the world population may be modelled based on observations from 1920 to 1992 as

$$y(t) = 2022.98 - 17.301t + 0.547t^2 + 0.555\text{x}10^{-6}t^3 + \xi,$$

(y in millions, and t=year–1900).

If two independent variables are used, a third-order model may be stated as

$$\begin{aligned} y = \quad & a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + b_1x_2 + b_2x_2^2 + b_3x_2^3 \\ & + c_1x_1x_2 + c_2x_1^2x_2 + c_3x_1x_2^2 + \xi, \qquad (8.2.4) \end{aligned}$$

where both the power terms and the cross product terms are considered.

Note that the models (8.2.3 - 8.2.4) are linear-in-the parameter models which is an advantage. The explosive tendency of square and high-power terms in the polynomial models can be a drawback. One of the ways to alleviate this problem is to consider different models based on some threshold criteria on the magnitude of some specific variables, similar to the threshold models (see Remark, Sec.8.4.3).

It is always desirable to use a polynomial with the *minimum* number of independent variables or regressors, to model the process representatively. There is no universal method available for selection of the optimum set of variables in all types of models. Subset selection (Secs.3.6.2–3.6.4) may be used to eliminate redundancy in

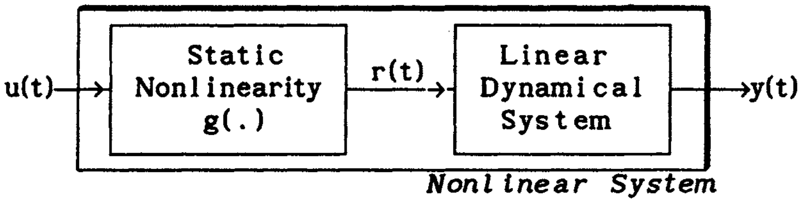


Figure 8.2.1 Hammerstein model of a nonlinear system.

the set of independent variables. For an example of selection of an optimal nonlinear AR model, see Sec.9.4.

Hammerstein model

This model (Fig.8.2.1) represents a nonlinear system by a static, zero-memory nonlinear system followed by a system with linear dynamics. If the nonlinearity $f_N(.)$ is known, the model can be considered to be a linear model with the input(s) $\{u(t)\}$ transformed to $\{f_N(u(t))\}$. If the nonlinearity is unknown, it may be expressed by a polynomial model; for example,

$$\begin{aligned} r(t) &= f_N(u(t)) \\ &= \alpha_1 u(t) + \alpha_2 u^2(t) + \dots + \alpha_m u^m(t). \end{aligned} \tag{8.2.5}$$

If the linear system is modelled as

$$A(q^{-1})y(k) = B(q^{-1})r(k-d), \tag{8.2.6}$$

where

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n},$$
$$B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_n q^{-n},$$

and d is the discrete time-delay between y(k) and r(k), the overall model response is given by

$$\begin{aligned} y(k) = (1-A(q^{-1}))y(k) + b_0(f_N(u(k-d))) + b_1(f_N(u(k-d-1))) \\ + \dots + b_n(f_N(u(k-d-n))). \end{aligned} \tag{8.2.7}$$

The parameters of (8.2.7) can be estimated as in case of a linear model. The model (8.2.7) is known as the Hammerstein model. The measurement y(k) in (8.2.6) may be noisy.

Wiener model

A basic Wiener model (Fig.8.2.2) consists of a system with

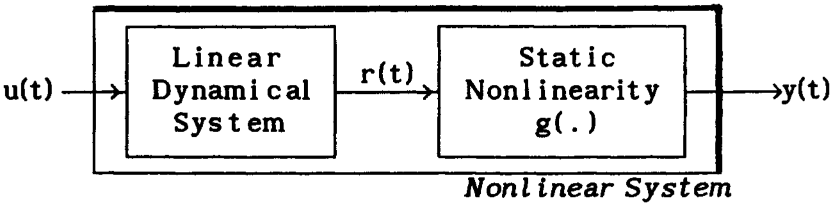


Figure 8.2.2 A basic Wiener model.

linear dynamics followed by a zero-memory system with static nonlinearity. A more complete Wiener model is shown in Fig. 8.2.3, which has three distinct sections. The input u(t) is fed to a series of linear dynamical systems, $h_i(t)$, which are orthogonal functions. The next section is a multi-input multi-output nonlinear system G(.) with no memory, the weighted sums of the outputs of which yield the single output y(t). Wiener had considered (Wiener, 1958) Laguerre functions for $h_i(t)$ and Hermite polynomials for G(.); excessive computational requirement of the generalized Wiener model led to the development of many simplified representations as discussed in Schetzen (1980).

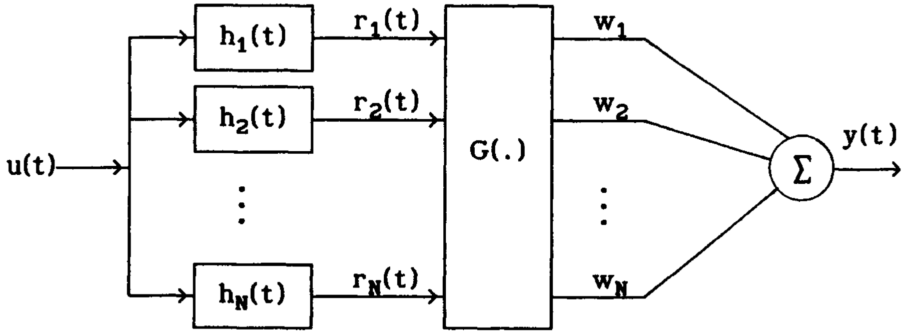


Figure 8.2.3 A general model of nonlinear systems of the Wiener class.

Nonlinear regression

Nonlinear regression is used to estimate the parameters of a nonlinear-in-parameters model. The estimation problem can be formulated as a nonlinear optimization problem which can be solved using standard algorithms, e.g., Gauss-Newton algorithm. Another approach can be linear extrapolation followed by iterative solution. Details of nonlinear regression are beyond the scope of this book; the reader may refer to Draper and Smith (1981), Bard (1974).

8.2.3 Nonlinear Transformation

The prime objective of nonlinear transformation is to expand the operating region, over which a simplified model of the underlying nonlinear process will be valid. In other words, the purpose of nonlinear transformation is to prepare the data for familiar methods of analysis, which cannot otherwise be used.

Nonlinear transformation can be performed both for time series data and input-output data. The usual practice is to transform either the dependent variable or the independent variable(s) or both. The graphical display of the dependent and the independent variables can help decide which variables need transformation. Subjective knowledge of the underlying process, if available, deserves primary consideration in deciding the type of transformation.

The two basic objectives of nonlinear transformation are linearization and variance equalization.

Linearization is performed when the time series shows a nonlinear trend or when a nonlinear relationship is observed between the dependent (or the response) variable and the independent (or the explanatory) variables. Some basic transformations z of y for linearization are as follows

$$z = y^n, \ldots, y^2, y, \surd y, \log y, 1/\surd y, 1/y, 1/y^2, \ldots, 1/y^n,$$

n being a positive integer. In general for multiplicative nonlinearity, logarithmic transformation is used.

Equalization of variance is necessary when the data show nonuniform spread or variability. For example, the variability in the Sunspot series (Appendix 8A) is much reduced when logarithmically transformed, as shown in Fig.8.2.4.

Some selected cases are mentioned below where particular transformations result in constant variance subject to some specific statistical relationships.

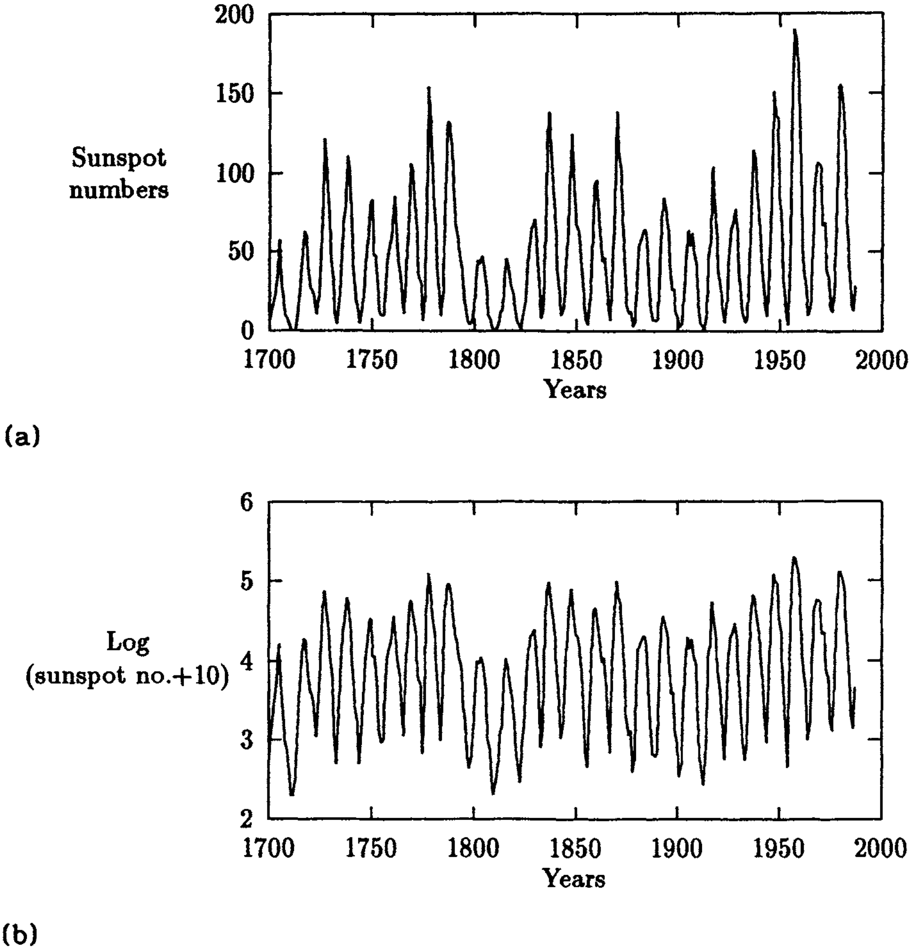


Figure 8.2.4 (a) The yearly averaged sunspot series $\{y(k)\}$, and (b) the transformed sunspot series $\{z(k)\}$ where $z(k) = \log_e(y(k)+10)$.

(a) *Square-root transformation* is used when variance is proportional to the expected value (i.e. $z = \sqrt{y}$, if $\mathrm{Var}(y) = E(y)$).

(b) *Logarithmic transformation* is used when standard deviation is proportional to expected value (for example, $z = \log y$, if $\sigma^2(y) = E(y)$).

(c) *Reciprocal transformation* is used when standard deviation is proportional to the square of the expected value (i.e. $z = 1/y$, if $\sigma^2(y) = (E(y))^2$).

One of the widely used and studied transformations is Box and Cox transformation (Box and Cox, 1964):

$$y(\lambda) = (y^\lambda - 1)/\lambda, \qquad \lambda \neq 0,$$

$$y(0) = \log y,$$

where the real unknown parameter λ is estimated by maximum likelihood method so the transformed data $\{y(\lambda)\}$ are independent and normally distributed with constant variance. See Atkinson (1985) for detailed discussions along with modifications. In such types of transformations where the parameters of transformation are estimated from the data, it is necessary to eliminate outliers, as otherwise the transformations can be adversely influenced.

Since saturation is inevitable with all real-life variables, sigmoidal type nonlinearity is also often used for transformation in nonlinear modelling. Sigmoidal nonlinearity is particularly popular with Neural Networks (see Sec.10.2).

When a linear model is used with the transformed data, it must be remembered that the disturbance term has also been transformed. Since linear regression requires

(a) additivity of the dependent terms,
(b) constancy of variance, and
(c) normal distribution of the data;

the degree of validity of these conditions needs to be reassessed when transformed data are used.

8.3 NONLINEAR PERIODICITY

Data series or signals with certain degrees of periodicity broadly fall into three categories: (i) periodic, (ii) quasiperiodic, (iii) chaotic series. In this section, the features of these classes of series are stated, and then their analysis is presented.

8.3.1 Periodic, Quasiperiodic and Chaotic Series

Periodic series

A data series is said to be *strictly periodic* if

$$x(t) = x(t+T),$$

for all t where the time period T(>0) remains unchanged. The process will be *nearly periodic* if

$$x(t) \simeq x(t+T),$$

Periodic series are discussed in Sec.7.7.1.

The frequency spectrum of a periodic series will comprise the fundamental component, $f = 1/T$, with the harmonic components at $f_n = n/T$, $n = 2,3,\ldots$ etc.

Remarks

(a) A periodic series shows a repeating pattern, the repetition frequency being the same as the periodicity.

(b) A periodic series may be generated from a linear combination of a number of sinusoids; the frequency component corresponding to the period length of the composite series will have the maximum magnitude.

Quasiperiodic series

A quasiperiodic series is a linear combination of two or more periodic series each of whose frequencies is a linear combination of a finite set of frequencies, at least two of which are incommensurate.

Remarks

(a) In the present context, the introduction of the following terms is in order.

(1) A number which can be expressed as a ratio of two integers is called a *rational number*.

(2) For a discrete-time series the normalized frequency (i.e. normalized with respect to the sampling frequency) has to be rational to be periodic.

(3) Two frequencies are called *incommensurate* if their ratio is not a rational number.

(b) If frequencies of at least two components are not incommensurate as stated above, the series can be locally quasiperiodic but globally periodic. □□

Consider the quasiperiodic series given by

$$x(t) = \sum_{j=1}^{m} g_j(t),$$

where f_j, the frequency of g_j, is a linear function of a set of linearly independent base frequencies, $r_1, r_2, \ldots, r_N$:

$$f_j = a_1 r_1 + a_2 r_2 + \ldots + a_N r_N,$$

$a_1, \ldots, a_N$ being integers.

The base frequencies are not uniquely defined but the number of base frequencies, N, is characteristic of a quasiperiodic series, which is also called N-periodic. A periodic series is a quasiperiodic series with N=1. The frequency spectrum of an N-periodic series shows N sets of harmonics at discrete frequencies. For g_1, the harmonics are at f_1, $2f_1$, $3f_1, \ldots$ etc. A quasiperiodic series can also be generated through nonlinear interaction of two or more periodic series.

The yearly averaged Sunspot activity series (Fig. 8.2.4(a)) is a typical example of a quasiperiodic series. The state-space diagram (say, x(k-1) vs. x(k) plot) of a quasiperiodic series is distinctly different from that of a periodic or a chaotic series, as shown in Example 8.3.2.

Chaotic series

A chaotic series shows the following distinguishing features:

(i) a continuous frequency spectrum, and

(ii) sensitivity to the initial condition.

A chaotic series is bounded in magnitude. It comprises a broad band of frequencies which manifests in noise like a continuous frequency spectrum. Due to the characteristic sensitivity to the initial condition, long-term prediction is not possible for a chaotic series, because irrespective of closely defined initial conditions the series and the predictor trajectories may increasingly diverge with time and eventually be uncorrelated with each other.

The oscillations in physiological series can be modelled as

$$x(k+1) - x(k) = \frac{\alpha x(k-\tau)}{1 + x^{\gamma}(k-\tau)} - \beta x(k), \tag{8.3.1}$$

which is the discrete-time representation of the Mackey Glass equation (Mackey and Glass, 1977). With suitable choices for the parameters α, β, γ and τ, (8.3.1) can model periodic, quasiperiodic or chaotic processes, as discussed in Example 8.3.2. A typical chaotic series generated from the Mackey Glass equation is shown in Fig.8.3.4(a).

8.3.2 Analysis using State-space Diagrams, SVD and FFT

The modelling and characterization of data series or processes with periodicity using SVD and FFT has been discussed in Sec.7.7 and Sec.2.5 respectively; the characteristic features can also be revealed through the state-space diagrams.

State-space diagrams

The x(k) vs. x(k-i), i≥1, diagrams of any sequence (or process) {x(k)} are referred to as the state-space diagrams in two dimensional space; state-space diagrams can also be formed in higher dimensional spaces.

A process with periodicity generates at least one closed contour in the state-space diagram. These closed contours are also called *limit cycles.* Some basic facts about these closed contours are as follows.

(a) If the sequence {x(k)} is strictly periodic, a repeating closed contour will be generated in the state-space diagram.

(b) If the sequence {x(k)} is sinusoidal, the closed contour will be elliptical in shape. If the sequence is periodic but contains more than one sinusoidal component, the closed contour will be repeating but will deviate from the elliptical shape.

(c) In real life, periodic processes are rarely *strictly periodic.* For example, consider the atmospheric CO_2 concentration series (Appendix 7C). The series has a trend, and is nearly periodic with yearly periodicity (see Figs. 8.3.1(a) and 8.3.1(b)). Such processes will generate closely placed closed contours (see Fig.8.3.1(b)). Note that the state-space diagram can be meaningful when the data series is *detrended,* as otherwise the contours move with a trend (as in Fig.8.3.1(a)) making the analysis difficult.

(d) Processes with periodicity are generally bounded. However, any tendency of unboundedness of a sequence is explicitly revealed in the state-space diagram, which shows gradually growing contours.

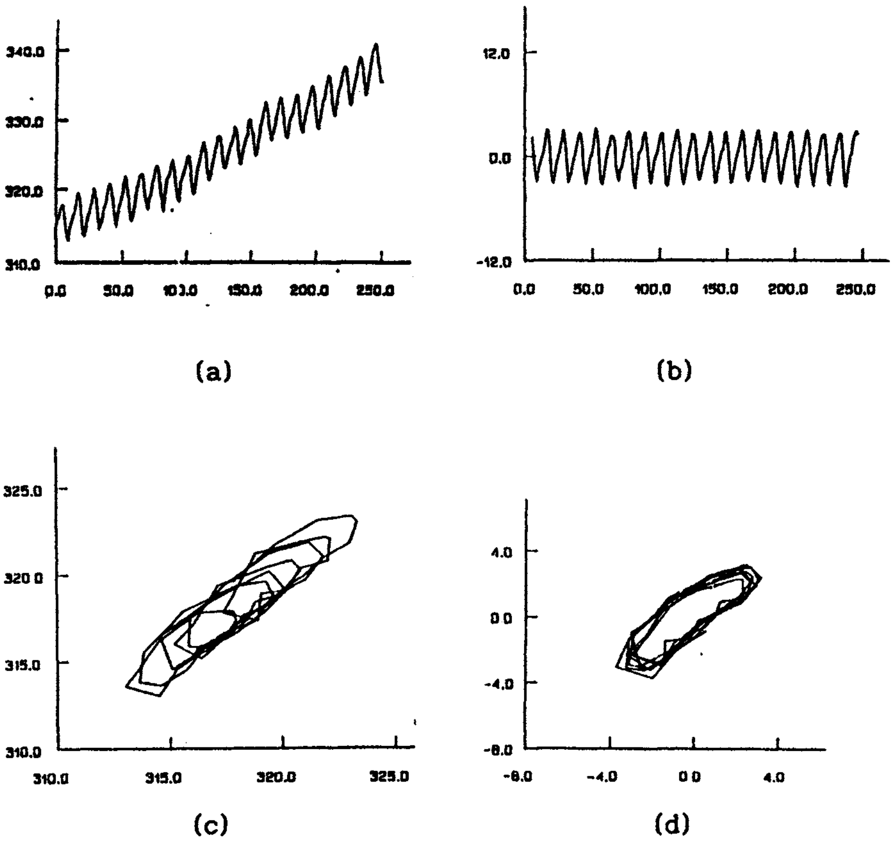


Figure 8.3.1 (a) The atmospheric CO_2 concentration series, showing monthly CO_2 conc. in ppm over the years 1959–80.
(b) the detrended CO_2 concentration series,
(c) state-space diagram for CO_2 concentration series,
(d) state-space diagram for the detrended CO_2 series.

Remark: The state-space diagram being a simple and concise representation, it may be possible to demonstrate the characteristic pattern or abnormality in any periodic or quasiperiodic process through state-space diagrams.

Comparative analysis

(a) Periodic processes:

The Fourier transform will produce components with respect to distinct *sinusoidal* components present in the signal. The SVD will produce *one* dominant singular value when the repeating or nearly repeating *patterns* (or periods) are aligned into consecutive rows of a matrix on which SVD is performed. As long as the sequence repeats or nearly repeats after a certain period length, there will be one dominant singular value. Thus if the repeating pattern comprises a number of sinusoidal components, there will be one singular value but as many Fourier components.

For a periodic process, the state-space diagram will show at least one closed contour per period. There are some basic differences between the state-space diagram analysis and the SVD based analysis as follows.

(i) There is no need for aligning the data in state-space analysis, as required in the case of SVD based analysis (see Sec.7.7 and Sec.11.2.1).

(ii) Within the periods, if there is any further explicit periodicity, the state-space diagram will show a smaller closed contour corresponding to each of such periods within the main period. Whether the smaller contours are discernible or not depends on the nature of the concerned periods. In the case of SVD, detection of smaller periods will be difficult.

(b) Quasiperiodic processes:
A quasiperiodic process is expected to have periods of varying length. Since for SVD analysis, the consecutive periods have to be aligned into the rows of a matrix, in the case of quasiperiodic processes either some interpolated data are to be used, or some data have to be sacrificed as discussed in Sec.11.2.1. There will be one dominant singular value, but the other singular values will not be insignificantly small, unlike the periodic case.

In the case of the state-space diagram analysis, if a quasiperiodic process is generated from a linear combination of p periodic processes, there will be at least q number of closed contours, where $1 \leq q \leq p$, if harmonic relations exist between the component processes. In case of nonharmonic components, the composite process will show at least p closed contours.

(c) Chaotic processes:
Chaotic processes can be analysed using the state-space diagram and the Fourier transform; no meaningful analysis is

possible using SVD through conventional approaches.

The state-space diagram of a chaotic process shows a large number of closed contours with no apparent order of arrangement. The contours will be nonintersecting in three dimensional or larger dimensional spaces. The difference between a quasiperiodic process and a chaotic process can be difficult to distinguish from the state-space diagram alone, and spectral analysis can be of use in such cases. The chaotic process will show the presence of an almost continuous, broad band, noise-like frequency spectrum, whereas the frequency spectrum for a quasiperiodic process will have one or more nearly repeating bands; in both cases the amplitude will decay for larger frequencies.

Remarks

(1) Quasiperiodic series tend to show gradual phase shift if a *periodic frame* is used to study its dynamics; the closed contours on state-space diagrams serve as such frame. They are less rigid than the row (or column) spaces in SVD and can pictorially exhibit the gradual phase shift by tracing successive contours that slip away from previous ones by incremental amounts.

(2) A periodic process loses its periodic property on being sampled at a rate which is not a rational submultiple of its periodicity. FFT will not reveal the periodicity with sharp peaks but state-space diagram will reveal the periodicity through a closed contour. A typical example is

$$x(t) = \sin(2\pi t/\sqrt{2}),$$

sampled at $t = 1,2,3,\ldots$

Example 8.3.2 Produce the state-space diagrams for the data series generated by the Mackey-Glass equation (8.3.1).

With reference to (8.3.1), consider the following cases.

Case 1: with $\alpha = 0.2$, $\beta = 0.1$, $\gamma = 10$ and $\tau = 6$, a nearly periodic sequence results.

Case 2: with $\tau = 17$, and the other parameters left as in Case 1, a relatively quasiperiodic sequence is generated.

Case 3: with $\alpha = 0.8$, $\beta = 0.4$, $\gamma = 16$ and $\tau = 95$, the sequence tends to be chaotic.

The state-space diagrams and the corresponding Fourier transforms for the above mentioned cases are shown in

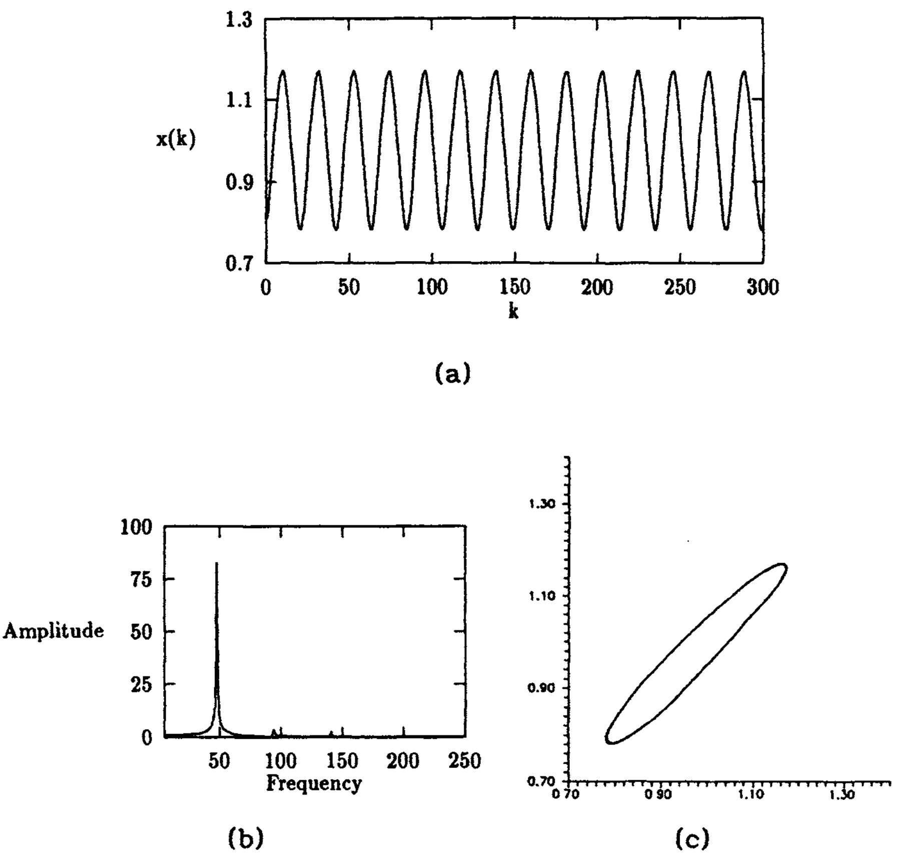


Figure 8.3.2 (a) The nearly periodic Mackey-Glass series (Case 1) with α=0.2, β=0.1, γ=10, and τ=6,
(b) the state-space diagram,
(c) the frequency spectrum showing primarily one frequency component.

Fig.8.3.2, Fig.8.3.3 and Fig.8.3.4 respectively.

Note that Case 1 shows primarily one frequency component being present. Case 2 is an example of a quasi-periodic series with primarily two different periodicities; the Fourier transform also confirms the presence of two prime bands of frequency components. The Fourier transform of Case 3 shows the presence of a wide range of frequency components, which confirms the process being chaotic.

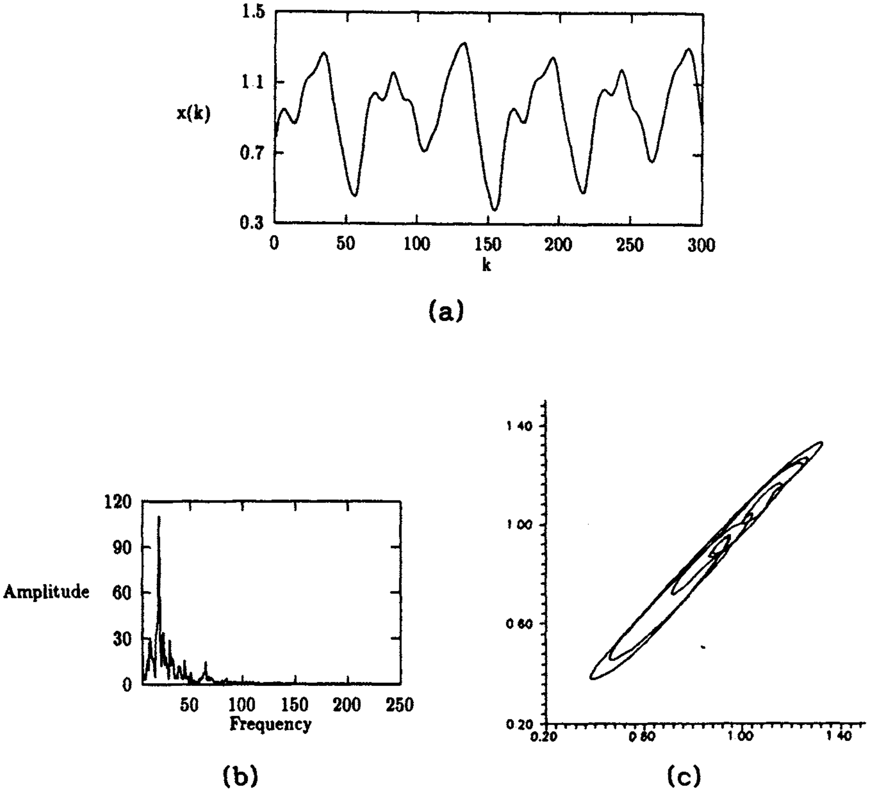
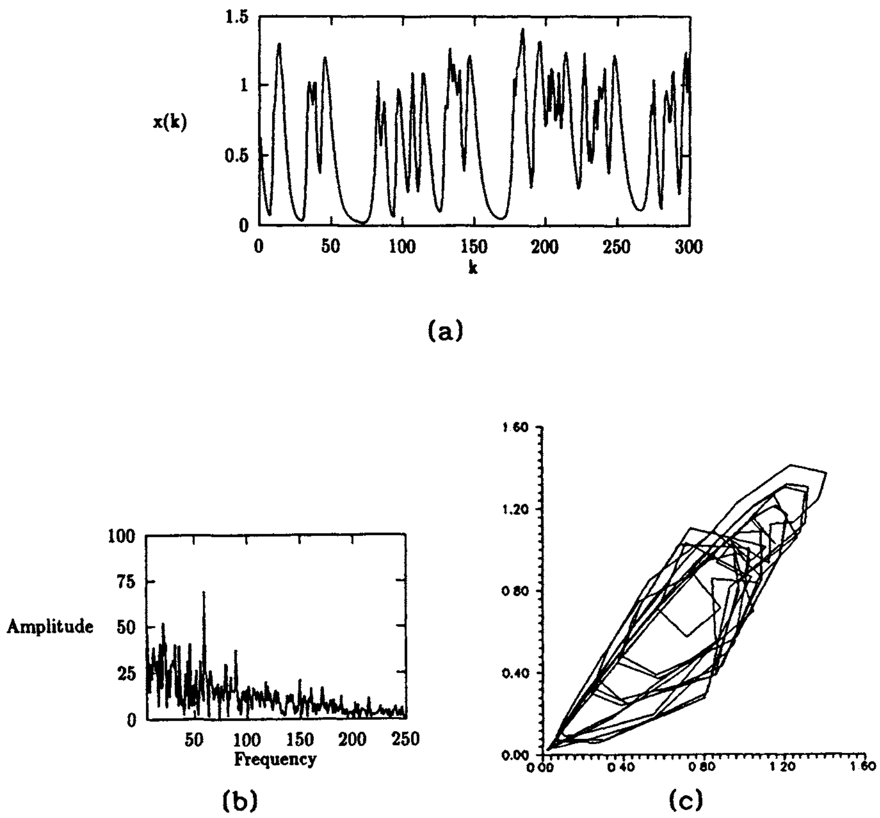


Figure 8.3.3 (a) The quasiperiodic MG series (Case 2) generated with α=0.2, β=0.1, γ=10, and τ=17,
(b) the state-space diagram,
(c) the frequency spectrum showing primarily two bands of frequencies.

8.4 SELECTED NONLINEAR MODELS

Some of the widely studied models which lend themselves to physical, biological and social sciences etc. are summarized here. A detailed study is beyond the scope of this book; only the underlying principles of natural representations of diverse dynamic processes are presented here.



(a)

(b)

(c)

Figure 8.3.4 (a) The chaotic Mackey-Glass series (Case 3) generated with α=0.2, β=0.1, γ=16, and τ=95, (b) the state-space diagram,
(c) the frequency spectrum showing relatively widely spread frequency components.

8.4.1 Bilinear Models

The nonlinearity in a bilinear model is present as multiplicative terms of two process variables:

$$y(k) + \sum_{i=1}^{n} a_i y(k-i) = \sum_{i=1}^{n} b_i w(k-i) + \sum_{h=1}^{p} \sum_{j=1}^{m} g_{hj} y(k-h) w(k-j), \qquad (8.4.1)$$

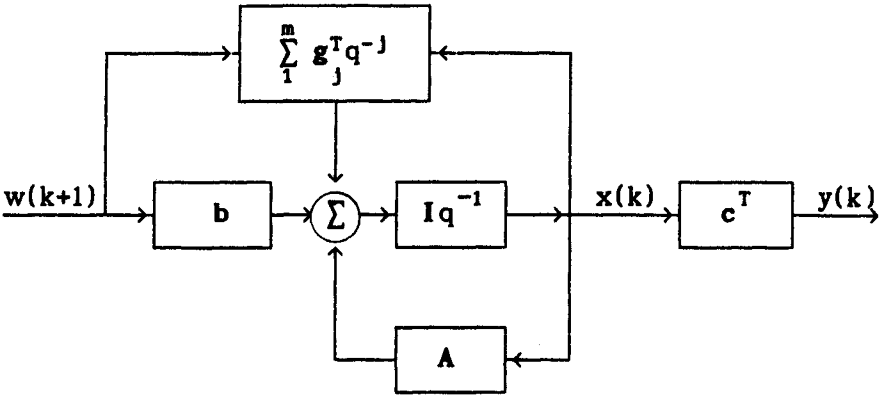


Figure 8.4.1 The state-space representation of the bilinear model (8.4.2 - 8.4.3).

where {y(k)} is the output sequence or a discrete-time series; {w(k)} is a sequence of input variables or equation error (or residual), which may also be a set of independent random variables; p, m and n are integers.

A state-space representation of (8.4.1) is given by

$$x(k+1) = Ax(k) + bw(k+1) + \sum_{j=0}^{m-1} g_j^T w(k-j)x(k), \qquad (8.4.2)$$

$$y(k) = c^T x(k), \qquad (8.4.3)$$

where

$$A = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \vdots \\ b_n \end{bmatrix},$$

$$c^T = [1\ 0\ 0\ \ldots\ 0], \qquad g_j^T = [g_{1j}\ g_{2j}\ \cdots\ g_{pj}],$$

for j = 1,2,...,m; and x is of size, max(n,p).

Thus the bilinear process given by (8.4.2-8.4.3) and shown in (Fig.8.4.1) is linear either in the state x or in the input (or equation error) w but not in both. The process is

called
(a) homogeneous in state, if $\mathbf{b} = 0$, or
(b) homogeneous in input (or equation error), if $\mathbf{A} = 0$, or
(c) strictly bilinear, if $\mathbf{A} = 0$ and $\mathbf{b} = 0$.

Since the linear and the nonlinear parts of a bilinear model are not orthogonal to each other, it cannot be said that the nonlinear part of the model represents the nonlinear part of the process.

Bilinear models can be formulated in continuous-time form or discrete-time form, which again can be stochastic or deterministic in nature.

To identify a bilinear time series model, Subba Rao and Gabr (1984, p.176) first consider the best subset AR model (see Sec.3.6.3); appropriate bilinear terms are then introduced through parameter estimation using the Newton-Raphson method, aiming at the best model in terms of minimum value of AIC.

Example
The sunspot activity series (Appendix 8A), modelled on 221 observations (i.e. for the years 1700 - 1920) is given by (Subba Rao and Gabr, 1984, p.197):

$$\begin{aligned} y(k) &- 1.5012y(k-1) + 0.767y(k-2) - 0.1152y(k-9) \\ &= 6.886 - 0.1458y(k-2)e(k-1) + 0.006312y(k-8)e(k-1) \\ &\quad - 0.007152y(k-1)e(k-3) - 0.006047y(k-4)e(k-3) \\ &\quad + 0.003619y(k-1)e(k-6) + 0.004334y(k-2)e(k-4) \\ &\quad + 0.001782y(k-3)e(k-2) + e(k), \end{aligned} \tag{8.4.4}$$

where $\{e(k)\}$ is a random noise sequence; the normalized AIC = 4.927 .

8.4.2 Threshold Models

If a nonlinear system displays varying dynamic characteristics over different operating regimes, it may be difficult to represent the process by a single model. Threshold models can be particularly useful in such cases. The basic idea is to express the nonlinearity in terms of a number of linear or nonlinear models, each being valid within specific operating regimes, defined by the thresholds.

Thresholds are numeric values. Threshold modelling defines different submodels which are valid subject to crossing of the threshold by a lagged observation, or a function of lagged observations. Each submodel is a partial description of the underlying nonlinear process. If the submodels are linear, a criterion like AIC is used for model order selection and the parameters are estimated using the least squares method or the maximum likelihood method. Each submodel is treated separately. AIC can also be used to decide the variable on which the threshold is to be set, as well as the appropriate value of the threshold. For detailed study on threshold modelling refer to Tong (1983).

Example

The sunspot activity series (Appendix 8A), over the years 1700 to 1920 is modelled by Tong and Lim (1980) as

$$y(k)=\begin{cases}10.544 + 1.692y(k-1) - 1.1592y(k-2) + 0.23674y(k-3) \\ \quad + 0.1503y(k-4) + e_1(k), \qquad \text{if} \quad y(k-3) \le 36.6 \\ 7.8041 + 0.7432y(k-1) - 0.0409y(k-2) - 0.202y(k-3) \\ \quad + 0.173y(k-4) - 0.2266y(k-5) + 0.0189y(k-6) \\ \quad + 0.1612y(k-7) - 0.256y(k-8) + 0.319y(k-9) \\ \quad - 0.3891y(k-10) + 0.4306y(k-11) - 0.03974y(k-12) \\ \qquad + e_2(k), \qquad \text{if} \quad y(k-3) \ge 36.6,\end{cases} \tag{8.4.5}$$

where e_1 and e_2 are white noise sequences. The normalized AIC for this model is 5.00.

Nonlinear threshold AR models have been studied by Ozaki (1981, 1985). The basic characteristic of such models is that at least one submodel is expressed as a nonlinear function having one or more parameters, which are themselves nonlinear functions of the amplitude. A generalized nonlinear AR model is given by

$$y(k) = \begin{cases} a_1y(k-1) +\ldots+ a_ny(k-n) + e_1(k), \quad \text{if } |y(k-1)| \ge C, \\ f_1(y(k-1)) + \ldots + f_n(y(k-1))y(k-n) + e_2(k), \\ \qquad\qquad \text{if } |y(k-1)| < C, \end{cases}$$

where

$$f_1(y(k-1)) = b_0 + b_1y(k-1) + \ldots + b_my(k-1)^m.$$

8.4.3 Exponential Models

The basic characteristic of an exponential autoregressive model (Ozaki, 1985) is one or more parameters being negative exponential functions of the variable. For example, consider a second order exponential AR model:

$$y(k) = (a_1 - b_1 e^{-y^2(k-1)})y(k-1) + (a_2 - b_2 e^{-y^2(k-1)})y(k-2) + e(k) \tag{8.4.6}$$

$$= \alpha_1 y(k-1) + \alpha_2 y(k-2) + e(k), \text{ (say).} \tag{8.4.7}$$

Thus parameter values of the exponential model are dependent on the magnitude of $y(k-1)$; if $y(k-1)$ is very small,

$$\alpha_1 \simeq a_1 - b_1, \text{ and } \alpha_2 \simeq a_2 - b_2,$$

and if $y(k-1)$ is very large,

$$\alpha_1 \simeq a_1 \text{ and } \alpha_2 \simeq a_2.$$

Thus the exponential function within the parameter acts as a smooth threshold.

The parameters a_i, b_i in (8.4.6) can be estimated by standard least squares estimation method as the model is linear-in-parameters. A suitable criterion may be used to determine the model order in the case of a generalized model:

$$y(k) = f_1(y(k-1))y(k-1) + f_2(y(k-1))y(k-2)$$
$$+ \ldots + f_n(y(k-1))y(k-n) + e(k),$$

where

$$f_i(y(k-1)) = a_i + b_i e^{-y^2(k-1)}.$$

An extended exponential model is represented by

$$y(k) = \{a_0 + (b_0 + b_1 y(k-1) + \ldots + b_m y^m(k-1))e^{-y^2(k-1)}\}y(k-1) + e(k),$$

which is equivalent to the exponential AR model for $m = 0$.

Remark

The linear and nonlinear threshold models and the exponential AR models belong to the category of models with variable amplitude dependent parameters. The purpose of constraining the parameters is stabilization, that is to arrest explosive tendencies of polynomial models.

8.5 CONCLUSIONS

There are many possibilities for modelling nonlinear processes, and there is a lot of flexibility as regards model structure selection and parameterization. However some of the basic problems are (a) there is a certain degree of heuristics associated with almost all modelling methods, and (b) the statistical procedures for the assessment of the convergence and the stability of the identified parameters and hence the quality of the model are not adequately developed. So proper identification can be difficult. Again, there can be different models for the same process, although no model can be said to be the best. As such, the model is not expected to be a reconstruction of the process, rather it is intended to serve as a set of operators on the identified set of inputs, producing similar output as expected from the process. The problem is that in real life the process output is usually contaminated with noise and other disturbances, whereas ideally the model output should follow the *true* output of the underlying representative process, which is unknown. So validity of a model should be examined carefully, for example through cross validation (i.e. validation against sets of representative data not used for modelling) or through prediction performance etc.

A summary of some of the well-studied nonlinear models namely, the bilinear model, the threshold model, and the exponential model has been presented in this chapter. A large class of processes can be represented by these models. The different models are not universally applicable. It is important that proper choice of the appropriate model is made depending on the application concerned, for which unfortunately there is no definite guideline.

REFERENCES

Remarks: Modelling and identification of nonlinear processes are surveyed in [3,7,19], and also covered in the texts [2,5]. The characteristics of quasiperiodic and chaotic processes are nicely explained in [13]; nonlinear periodicity is treated in [9]. The problems of using SVD for modelling chaotic processes is studied in [14]. Some early references on nonlinear transformation are [4,20], and it also features in the book [1]. Wiener approach to nonlinear modelling is treated in [14, 21]. Bilinear modelling is treated in [6,10,16]. Threshold models are studied in [18,19,11]. Exponential models are discussed in [11,12].

Modelling and prediction of quasiperiodic processes are also treated in Chapters 9, 10, and 11 of this book.

[1] Atkinson, A.C. (1985): *Plots, Transformations and Regression*, Oxford Univ. Press, Oxford.

[2] Bard, Y. (1974): *Nonlinear Parameter Estimation*, Academic Press, New York.

[3] Billings, S.A. (1980): 'Identification of nonlinear systems - a survey', *Proc. IEE*, Pt.D, 127(6), 272-284.

[4] Box, G.E.P, and D.R. Cox. (1964): 'An analysis of transformations', *Journal of Royal Stat. Soc.*, Ser.B, 26, 211-252.

[5] Draper, N. R., and H. Smith (1981): *Applied Regression Analysis*, 2nd ed., Wiley, New York.

[6] Granger, C.W.J., and A.P. Anderson (1978): *An Introduction to Bilinear Time Series Models*, Vanden Hoeck and Rupcecht, Gottingen.

[7] Haber, R., and H. Unbehauen (1990): 'Structure identification of nonlinear dynamic systems - a survey on input/output approaches', *Automatica*, 26(4), 651-677.

[8] Mackey, M.C., and L. Glass (1977): 'Oscillations and chaos in physiological control systems', *Science*, 197, No. 4300, 287-289.

[9] Minorsky, N.J. (1962): *Nonlinear Oscillations*, D. Van Nostrand, Princeton, N.J.

[10] Mohler, R.R., and W.J. Kolodziej. (1980): 'An overview of bilinear system theory and applications', *IEEE Trans. Syst. Man and Cyber.*, SMC-10(10), 683-688.

[11] Ozaki, T. (1981): 'Nonlinear threshold autoregressive models for nonlinear random vibrations', *Journal of Applied Probability*, 18(2), 443-451.

[12] Ozaki, T. (1985): 'Nonlinear time series models and dynamical systems', in *Handbook of Statistics*: Vol.5, 'Time series in the time domain', Eds. E.J. Hannan, P.R. Krishnaiah and M.M. Rao, 25-83, North-Holland, Amsterdam.

[13] Parker, T.S., and L.O. Chua (1987): 'Chaos: a tutorial for engineers', *Proc. IEEE*, 75(8), 982-1008.

[14] Paluš, M., and I. Dvorák (1992): 'Singular-value decomposition in attractor reconstruction: pitfalls and precautions', *Physica D*, 221-224.

[15] Schetzen, M. (1980): *The Volterra and Wiener Theories of Nonlinear Systems*, Wiley, New York.

[16] Subba Rao, T., and M.M. Gabr (1984): *Introduction to Bispectral Analysis and Bilinear Time Series Models*, Lec. Notes in statistics, **24**, Springer-Verlag, Berlin.

[17] Tong, H., and K.S. Lim, (1980): 'Threshold autoregression, limit cycles and cyclic data', *Journal of Royal Statistical Society*, Ser. B, **42**, 245-292.

[18] Tong, H. (1983): *Threshold models in non-linear time series analysis*, Lecture Notes in Statistics, No.21, Springer-Verlag, New York.

[19] Tong, H. (1987): 'Nonlinear time series models of regularly sampled data: a review', Proc. Ist World Congr. Bernoulli Society, Tashkent, Vol.2, 355-367, VNU Science Press, Amsterdam.

[20] Tukey, J.W. (1957): 'On the comparative anatomy of transformations', *Annals of Math. Statistics*, **28**, 602-632.

[21] Wiener, N. (1958): *Nonlinear Problems in Random Theory*, John Wiley and Sons, New York.

CHAPTER 9

MODELLING OF NONLINEAR PROCESSES USING GMDH

Nonlinear processes can be modelled using hierarchical stages of simple nonlinearity, where each building block is represented by a linear-in-the-parameter model.

9.1 INTRODUCTION

The identification and modelling of processes with nonlinearity can often be a difficult task mainly because of the following reasons:

(a) unknown or partially known structures,

(b) large dimensions with many variables having nonlinear interrelations, and

(c) availability of limited operational data, etc.

The traditional functional series approaches of Volterra and Wiener require prohibitively intensive computation even for low order representations; a large parameter set as well as long data sequences are required for estimation. The Group Method of Data Handling (GMDH) offers a powerful alternative, which has been successfully used in diverse areas (Farlow, 1984).

GMDH, which was developed by Ivakhnenko (1970), has a multilayer hierarchical structure. The basic building block is a two-input one-output submodel, represented (typically) by a linear-in-the-parameter quadratic polynomial. A bank of submodels forms a layer, and a bank of layers forms the model. The submodel outputs of one layer undergo a selection process for assessment of the richness of information contained, and the selected outputs are then forwarded as inputs to the following layer and so on. Thus a process with a high order of nonlinearity can be configured into a multilayer model comprising basic building blocks.

A crude single stage approximation of GMDH is to consider a set of all possible variables and their nonlinear transforms which can be related with the output through a linear-in-the-parameter model. A subset selection is performed on the input data set, and the selected data set is used for parameter estimation.

The organization of this chapter is as follows. The GMDH architecture is introduced in Sec.9.2. The two key issues on which the success of GMDH largely depends are the selection of candidate inputs for the layers and the selection of submodel structures, which are treated in Sec.9.3. Sec.9.4 features one typical application of GMDH involving a multi-input single-output environmental process. Sec.9.5 presents a single-layer nonlinear model which incorporates nonlinearization of input variables and identification using orthogonal transformation. All the models discussed in this chapter are linear-in-the-parameter models.

9.2 THE GMDH ARCHITECTURE

9.2.1 Multinomial Representation of Nonlinearity

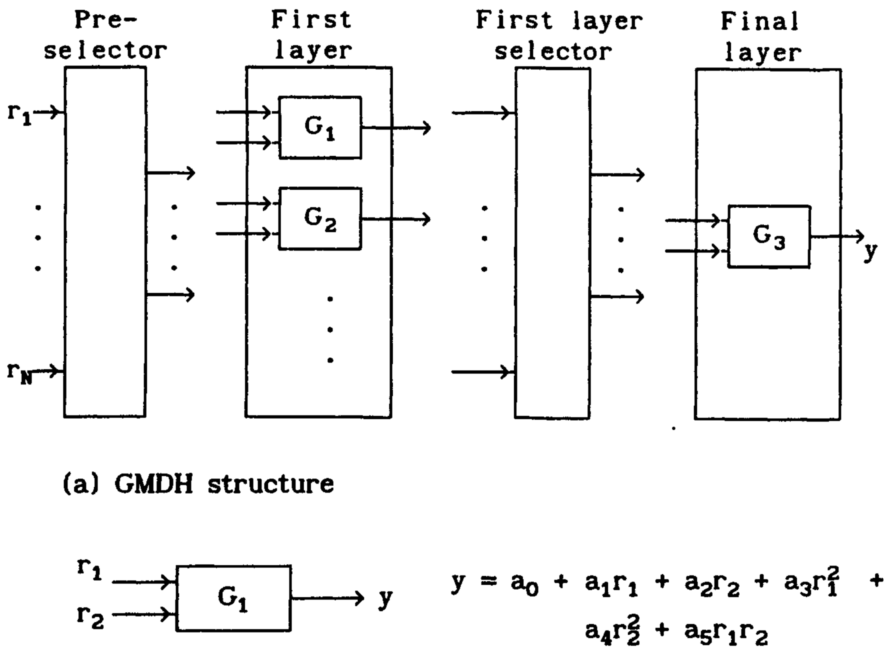
Consider a multi-input single-output nonlinear process. The output (y) can be, in general, expressed as a multinomial function (also called *Kolmogorov-Gabor Polynomial*) connecting all possible inputs and their combinations:

$$y = a_o + \sum_{i=1}^{N} a_i r_i + \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} r_i r_j + \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} a_{ijk} r_i r_j r_k + \cdots \tag{9.2.1}$$

where r_i, r_j etc. may represent known individual inputs, the square or higher powers of the same inputs; the time-lagged or past values of individual inputs or the output may also be treated as input to the model. a_i, a_{ij}, a_{ijk} are the unknown parameters of the system. Equation (9.2.1) is the discrete analogue of the Volterra series (Schetzen, 1980). The conventional methods for the estimation of the parameters of (9.2.1) will require large sets of data and prohibitively intensive computation. GMDH offers a simplified approach to this problem.

9.2.2 Structural Layout of GMDH

The GMDH model of (9.2.1) is shown in Fig.9.2.1. It consists



(a) GMDH structure

$$y = a_0 + a_1 r_1 + a_2 r_2 + a_3 r_1^2 + a_4 r_2^2 + a_5 r_1 r_2$$

(b) The generic submodel description

Figure 9.2.1 The schematic structural layout of GMDH

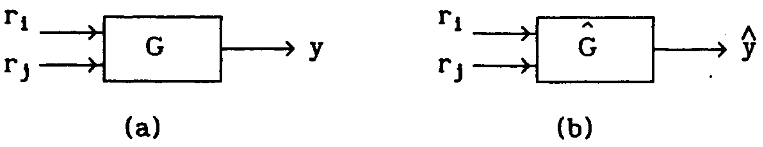


Figure 9.2.2 Modelling approach for submodels in GMDH:
(a) Parameter estimation scheme,
(b) Pseudo-output ($\hat{y}$) generation scheme.

of a series of layers, each having a number of submodels. A *submodel* (Fig.9.2.2) has one or two inputs and one output, and is modelled usually by a polynomial of maximum order two; for the parameter estimation of a submodel, the final output (y) is presumed to be the submodel output, and the estimated output is used as the submodel output for subsequent modelling operations. All sensible combinations of candidate inputs to a layer are accommodated as inputs to the submodels of the layer. Each layer is followed by a stage, called the *layer selector*. The function of the layer selector is to select those submodel outputs which contain maximum yet mutually independent information about the final output. These outputs, which are also called intermediate variables or pseudo-outputs, are passed on as inputs to the next layer. The outputs of subsequent stages are expected to bear stronger correlation with the final output.

There is no constraint on the number of submodels in a layer or the number of layers in the overall model. Due to the generation of intermediate variables with increasingly richer information, the number of submodels naturally drops in subsequent layers. At the final layer, the output (y) may be expressed as a linear function of the inputs to that layer.

9.3 GMDH: DESIGN AND VALIDATION OF MODELS

The design of GMDH models mainly involves the following problems:

(a) input selection for the first and subsequent layers,
(b) structure selection, including determination of suitable variables for the submodels and estimation of the parameters of the submodels in different layers, and
(c) the layer termination, which is effected when modelling error cannot be reduced any further by additional layers.

Input selection and layer selection

Input selection or *preselection* refers to selection of the inputs to the first layer, whereas layer selection refers to the selection of inputs to second or any subsequent layer; the basic purpose is the same for the input selector and the layer selector.

The set of candidate inputs **x**(k) to the first layer may include all the individual inputs (u_1, u_2 etc.), the time delayed inputs ($u_1(k-1)$, $u_1(k-2)$ etc.), and the time delayed outputs ($y(k-1)$, $y(k-2)$ etc.); in other words all variables which may be strongly or weakly correlated with the output are considered:

$$x(k) = \{u_1(k), u_1(k-1), \ldots, u_1(k-n), u_2(k), \ldots, u_2(k-n), \ldots,$$
$$u_N(k-1), \ldots, u_N(k-n), y(k-1), y(k-2), \ldots, y(k-n)\}. \quad (9.3.1)$$

In the selection of the input variables, their relevance with respect to the output should be given due consideration. If necessary, instead of the actual values, nonlinearly transformed values (Sec. 8.2.3) of the input variables may be used. Those variables which are relatively strongly correlated with the output are selected.

For *layer selection* the indices of submodel structure validation like the cumulative mean square error (MSE) cross validation for each submodel may be considered. Submodel outputs with relatively low values of cumulative MSE only are selected and the rest are dropped.

Collinearity between the candidate inputs both at the preselector as well as at the layer selector is undesirable. The chances of collinearity is particularly high at higher layers of GMDH. The collinearity can be detected using SVD, and to eliminate collinearity subset selection using QRcp factorization (discussed in Sec.3.6.2) may be used.

Once *distinct* collinearity among the candidate set of variables is eliminated, the significant set of variables may be subsequently chosen successively using the modified QRcp factorization as discussed in Sec.3.6.4.

Structure selection and validation

Structure selection and validation are closely related concepts, which in the present context relate to the appropriate modelling of the submodels. This subject is discussed in Sec.3.6; a brief outline follows.

The objective is to determine the model order and parameter values for each submodel, when a set of input and output data are available. It is understood that the models are linear in the parameters. When the model is underparameterized, that is the model order is too low, the input-output data will show high model-misfit with the estimated parameters. When the model is overparameterized, the model misfit may be low but overfitting of the data will result; there will be a tendency to model both the information and the noise in the data, and the validity of the model for sets of data not used for modelling will be poor. The two common approaches for structure selection and validation are

(i) to use C_p statistic (Sec.3.6.4),
(ii) cross validation (Sec.3.6.5).

Summary of GMDH

GMDH is performed through the following steps:

(1) The set of all the available inputs, **x** in (9.3.1), is formed. Preselection is performed using correlation analysis between each variable and the final output, or through subset selection.
(2) The modelling of the submodels is performed.
(3) The layer selection is performed using the information criterion index or the cross validation index.
(4) Steps (2) and (3) are repeated sequentially until an additional layer does not improve the information criterion index or the cross validation index.
(5) The final output is expressed as a linear function of the outputs of the last stage.

9.4 MODELLING THE COD PROCESS IN OSAKA BAY

The chemical oxygen demand (COD) can be considered to be an index of water pollution in the sea. COD concentration is monitored at a number of stations in the Osaka bay along with water temperature, transparency and dissolved oxygen concentration. Altogether 84 sets of monthly data are available (see Appendix 9), for the period 1976 to 1983; 54 sets of data are used as the training set and 15 sets of data are used as the testing set. The prediction performance of the model is tested on the last 15 sets of data.

Inputs and outputs

No. of inputs:	3	
Input variables:	x_1	water temperature
	x_2	water transperancy
	x_3	dissolved oxygen concentration
Output variable:	y	COD concentration

Preselection

Based on the process information, all the input variables are considered as inputs to the first layer, bypassing the stage of preselection. The submodels of the first layer are modelled as follows.

First layer

The submodels of the first layer are modelled as

$$u_1 = 5.1286 + 0.0162x_1 - 1.0918x_2 + 0.0060x_1^2 + 0.2864x_2^2 - 0.0688x_1x_2,$$

$$u_2 = -0.3446 - 1.7541x_2 + 2.1154x_3 + 0.4878x_2^2 - 0.1018x_3^2 - 0.2533x_2x_3,$$

$$u_3 = 6.8907 - 0.4275x_1 - 0.5813x_3 - 0.0097x_1^2 + 0.0292x_3^2 + 0.0290x_1x_3,$$

where u_1, u_2, and u_3 are the pseudo-outputs of the first layer.

Layer selection

The mean square error for the pseudo-outputs of the first layer on the validation data set, is given in the second column of Table 9.4.1. The MSE for the pseudo-outputs u_1, u_2, and u_3 are found to be close, and further rank tests reveal these variables not to be collinear. Hence all these pseudo-outputs of the first layer are considered as inputs to the next layer.

Second layer

The submodels of the second layer are modelled as follows.

$$v_1 = -0.2737 + 1.4732u_1 - 0.5016u_2 - 0.6828u_1^2 - 0.5256u_2^2 + 1.2510u_1u_2,$$

$$v_2 = -1.0212 + 0.3353u_2 + 0.9661u_3 - 0.2140u_2^2 - 0.2358u_3^2 + 0.4507u_2u_3,$$

$$v_3 = -1.6663 + 0.1894u_1 + 1.4523u_3 - 0.2955u_1^2 - 0.4513u_3^2 + 0.7057u_1u_3,$$

where v_1, v_2, and v_3 are the pseudo-outputs of the second layer.

Layer selection and layer termination

The MSE on the validation data set, as given in the second column of Table 9.4.1, shows v_2 producing minimum error among the second layer outputs. The modelling procedure is extended to the third layer, where the estimates obtained are as follows.

$$\begin{aligned} w_1 &= -0.6435 + 0.0328v_1 + 1.2295v_2 - 0.0378v_1^2 - 0.1635v_2^2 \\ &\quad - 0.1794v_1v_2, \\ w_2 &= -0.2281 + 1.7239v_2 - 0.6083v_3 - 1.1046v_2^2 - 0.7283v_3^2 \\ &\quad + 1.8224v_2v_3, \\ w_3 &= 0.2557 + 2.5392v_1 - 1.6596v_3 - 0.2919v_1^2 + 0.3211v_3^2 \\ &\quad - 0.0184v_1v_3, \end{aligned}$$

where w_1, w_2, and w_3 are the pseudo-outputs of the prospective third layer; the corresponding MSE are shown in the second column of Table 9.4.1, which confirms that v_2 leads to the lower error model, compared to the third layer outputs. So the model is terminated at the second layer.

Final model and the result

Using *all of the available* (the first 69) *data sets* the parameters of the submodels concerned are reestimated as follows.

$$\begin{aligned} u_1 &= 5.3744 - 0.1998x_1 - 0.5577x_2 + 0.0099x_1^2 + 0.1727x_2^2 \\ &\quad - 0.0336x_1x_2, \\ u_2 &= 0.5658 - 1.2108x_2 + 1.5735x_3 + 0.4478x_2^2 - 0.0605x_3^2 \\ &\quad - 0.2778x_2x_3, \\ v_2 &= -0.3130 + 0.1275u_1 + 0.8515u_2 - 0.1716u_1^2 - 0.1958u_2^2 \\ &\quad + 0.4029u_1u_2. \end{aligned}$$

The validation is assessed by computing the MSE for the estimated model on the last 15 sets of data (which are not used for modelling in any way). The results for all the submodels are presented in the third column of Table 9.4.1; once again v_2 is found to produce the minimum MSE (=1.0006) on the validation data set (i.e. 70th to 84th data set).

Remarks:

(1) The modelling and the validation data sets could be

Table 9.4.1 COD process model: MSE for GMDH-submodels

Pseudo-outputs	Model fit on first 54 data sets	Validation fit on 55th to 69th data sets	Validation fit on 70th to 84th data sets
u_1	0.9749	4.7535	2.4659
u_2	1.1908	2.0994	1.1236
u_3	0.8902	1.7354	1.4076
v_1	0.7973	20.6531	1.3732
v_2	0.7299	1.9124	1.0006
v_3	0.6890	2.7979	1.6492
w_1	0.7099	10.5616	1.0789
w_2	0.6715	2.8795	1.0904
w_3	0.6633	474.6286	1.3800

selected randomly over the available data sets, since no historic (i.e. time delayed) informations are used here.

(2) In the present case, the individual submodels have not been optimised. C_p statistic (discussed in Sec.3.6.4) coupled with QR factorization based subset selection may be used for this purpose.

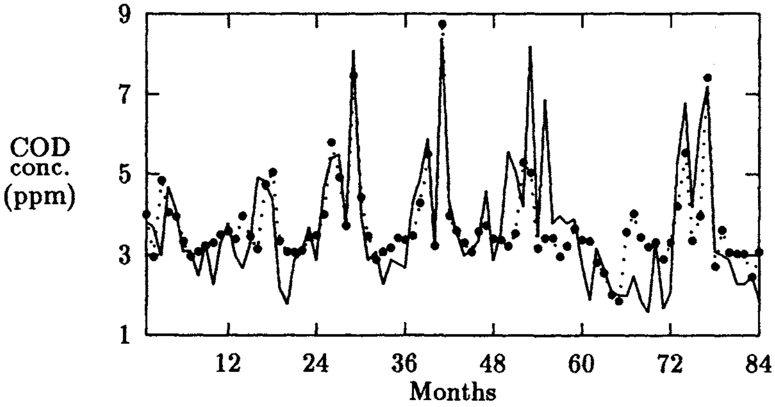


Figure 9.4.1 Estimation of the Chemical Oxygen Demand (COD) in the Osaka bay using the two layer GMDH model.

9.5 A SINGLE LAYER NONLINEAR MODEL BASED ON ORTHOGONAL TRANSFORMATION

The two basic features of GMDH are nonlinearization and multilayer identification of linear-in-the parameter models. This section discusses a simplified single-stage method, where the linear as well as selected set of nonlinearly transformed variables are considered. The objective is to produce the best model in an information criterion sense. The method is presented as an illustrative example.

Example 9.5.1 Modelling the yearly averaged sunspot series

A method for the best subset AR modelling is presented in Sec.3.6.3. Here a linear-in-the-parameter nonlinear model is considered.

For the sunspot series, the best subset AR model based on 221 data points over the years 1700 to 1920 given by (3.6.4) (see Example 3.6.3(1)) has three independent variables y(k-1), y(k-2) and y(k-9), while the candidate set of variables were {y(k-1), y(k-2), y(k-3),..., y(k-9)}. Here in addition to these nine variables, the candidate set is assumed to comprise all the quadratic terms involving y(k-1), y(k-2), y(k-9). Thus the candidate set becomes

$$\begin{aligned}&\{y(k-1),\ y(k-2),\ y(k-3),\dots,\ y(k-9),\\ &\ y^2(k-1),\ y^2(k-2),\ y^2(k-9),\quad \text{and}\\ &\ y(k-1)y(k-2),\ y(k-2)y(k-9),\ y(k-1)y(k-9)\}.\end{aligned}$$

The same procedure as detailed in Sec.3.6.3 is followed, which involves the use of subset selection and consideration of the information criteria AIC and SIC. The best model is obtained as

$$\begin{aligned}y(k) = {} & 1.3162y(k-1) - 0.4406y(k-2) - 0.1947y(k-3)\\ & + 0.1339y(k-8) + 0.0105y^2(k-2) - 0.0130y(k-1)y(k-2),\end{aligned}$$

with AIC = 5.0656, and SIC = 5.1760.

Here the best model is obtained through an exhaustive search on the selected subset with pseudorank 12, for which both AIC and SIC work out to be minimum.

Remark
It may be noted that subset selection can take care of the closeness to collinearity or the linear dependence among the input variables (or their nonlinearized variants) but their degree of correlation with the output (either singly or jointly) is not taken into account. On the other hand, the information criterion, cannot detect collinearity between the input variables (or regressors) but does take into account the correlation between the input(s) and the output.

9.6 CONCLUSIONS

GMDH is a powerful method for modelling nonlinear time series or input-output processes with limited data.

The main strength of GMDH is in the breaking up of a complicated problem of identification of a process with multiple inputs and single output into a number of simpler problems of identification of submodels with one or two inputs and one output. The number of parameters of a submodel being only a few (typically five), only a small data set is required for the identification of the complex process. The varied implementations of GMDH reported in the literature underline the application potential of this method.

It is expected that the use of orthogonal transformation and subset selection in the implementation of GMDH can lead to reduction in the heuristics and redundancy in the model structure, although more work needs to be done in this direction.

REFERENCES

Remarks: GMDH is introduced by Ivakhnenko in [6,7]; both theory and applications of GMDH arc widely reported in the issues of *Soviet Automatic Control*, published from 1969 onwards. The general statistical aspects of multilayer learning models are discussed in [1]. Some early applications of GMDH are reported in [3,5], and an early review appears in [10]. Applications are further reported in [8,15]. The book [4] is devoted to GMDH; it provides a detailed review of the subject along with varied application examples. The model structure selection and model validation are problems central to GMDH which are briefly addressed in Sec.3.6; many texts cover this subject in detail, e.g., [2,9,14].

[1] Barron, A.R., and R.L. Barron (1988): 'Statistical learning networks: a unifying view', Computing Science and Statistics: 1988 Proc. of the 20th Symposium on the Interface, 192-203.

[2] Draper, N.R., and H. Smith (1981): *Applied Regression Analysis*, 2nd. edn., Wiley, New York.

[3] Duffy, J.J., and M.A. Franklin (1975): 'A learning identification algorithm and its application to an environmental system', *IEEE Trans. Systems, Man, and Cybernetics*, SMC-5, (2), 226-240.

[4] Farlow, S.J. (Ed.) (1984): *Self-organizing Methods in Modelling: GMDH Type Algorithms*, Mercel Dekker, New York.

[5] Ikeda, S., M. Ochiai, and Y. Sawaragi (1976): 'Sequential GMDH algorithm and its application to river flow prediction', *IEEE Trans. Systems, Man, and Cybernetics*, SMC-6, (7), 473-479.

[6] Ivakhnenko, A.G. (1970): 'Heuristic self-organization in problems of engineering cybernetics', *Automatica*, 6, 207-219.

[7] Ivakhnenko, A.G. (1971): 'Polynomial theory of complex systems', *IEEE Trans. Systems, Man, and Cybernetics*, SMC-1, No.4, 364-378.

[8] Kortmann, M., and H. Unbehauen (1988): 'Two algorithms for model structure determination of nonlinear dynamic systems with applications to industrial processes', Preprints 8th IFAC Symp. on Identification and System Parameter Estimation, Beijing, Vol.2, 939-946.

[9] Ljung, L. (1987): *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, N.J.

[10] Mehra, R.K. (1977): 'Group Method of Data Handling (GMDH): review and experience', *IEEE Conf. on Decision and Control*, New Orleans, 29-34.

[11] Schetzen, M. (1980): *The Voltera and Wiener Theories of Nonlinear Systems*, Wiley, New York.

[12] Fujita, S., and H. Koi (1984) 'Application of GMDH to Environmental System Modelling and Management' in *Self-Organizing Methods in Modelling*, S.J. Farlow (Ed.),

Marcel Dekker, New York, 257–275.

[13] Shin-Ichi, Fujita (1991): Private communication.

[14] Söderström, T., and P. Stoica (1988): *System Identification*, Prentice-Hall, New York.

[15] Tamura, H., and T. Kondo (1980): 'Heuristics free group method of data handling algorithm of generating optimal partial polynomials with application to air pollution prediction', *International Journal of Systems Science*, 11(9), 1095–1111.

CHAPTER 10

MODELLING AND PREDICTION OF NONLINEAR PROCESSES USING NEURAL NETWORKS

Nonlinear series with or without periodicity as well as nonlinear input-output processes can be modelled using neural networks.

10.1 INTRODUCTION

Neural networks offer some of the most versatile ways of modelling nonlinear processes of a diverse nature. A neural network attempts to mimic the functioning of the brain in a crude but simplistic manner. The nonlinear relationship between the input(s) and the output(s) is modelled using a number of basic blocks, called *neurons* or *nodes*. The nodes are interconnected and are usually arranged in multiple *layers*. Each internodal *link* or *interconnection* is weighted. At each node, the weighted inputs (from other nodes or from external inputs to the network) are summed together with an external bias known as the *threshold*, and the result is passed through a nonlinear function (also known as the activation function), which forms the output of the node. The nonlinearity associated with each node remains fixed.

The weights on the interconnections are estimated iteratively by a nonlinear optimization method using known sets of input and output data; such adaptation of the network is referred to as the training or the learning of the network. The underlying idea is the biological nervous system-like performance of the network in learning complex processes. The main characteristic features are the network architecture, the nonlinearity associated with the nodes and the training method used. The designs are not unique, and there may be heuristics associated with the specification of the network structure.

Various configurations are possible for the neural networks for different applications. The present study is confined to the feedforward network, which is one of the most popular architectures. Modelling and prediction of nearly periodic and quasiperiodic time series as well as multi-input single-output processes are studied.

There is little difference between the configuring and training of a feedforward network and the conventional identification problem in the case of an input-output process. So it is desirable that the network is parsimoniously designed. The optimization of the size of the network (in terms of the optimum number of nodes and links) using singular value decomposition (SVD) and subset selection has been explored. Two methods used are:
(i) optimization through the selection of optimal set of time-domain inputs, and the optimal set of links and nodes within the neural network,
(ii) optimization through orthogonalization of the data prior to use in the neural network in the case of series with periodicity.

The organization of this chapter is as follows. The basic features of neural networks are presented in Sec.10.2. The multilayer perceptron structure is introduced in Sec. 10.3, and the backpropagation algorithm for the adaptation of the weights on the links and the thresholds is discussed. The design of feedforward neural networks of optimal size is treated in Sec.10.4; the modelling of both time series as well as complex input-output processes is considered. Section 10.5 is devoted to the study of neural networks operating with transformed input output data, modelling nearly periodic or quasiperiodic processes. Finally, Sec.10.6 introduces an SVD based method for the convergence assessment during the training of the neural network, which can be used as an alternative to the conventional approach based on the output error.

10.2 BASICS OF NEURAL NETWORKS

A node

A node, which is the basic component of a neural network, is designed to mimic the understanding of the functionality of a neuron in the human brain. The inputs to a node (Fig.10.2.1) are the available measurements or the outputs from other nodes. Each input is treated as a connection or a link with which a weight is associated. Each node (Y) is

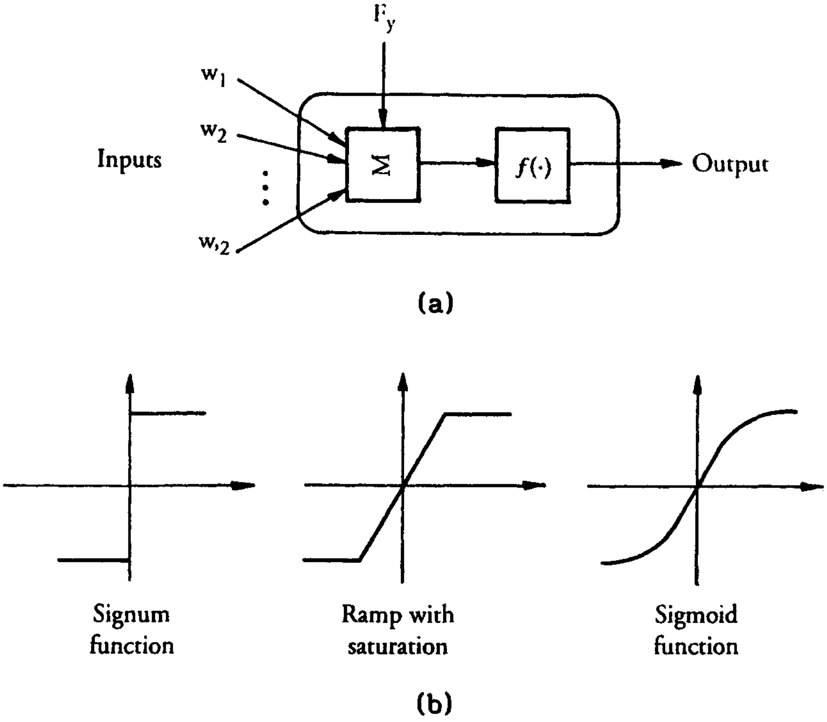


Figure 10.2.1 (a) The structure of a node, and (b) typical activation functions.

characterized by a nonlinear function (f(.)) and an additive threshold value (F_y). The node sums the weighted inputs and the threshold value, and passes the result through its characteristic nonlinearity to produce the output. The threshold is used as an offset.

The three common types of nodal nonlinearities are as follows:

(a) Signum function

$$f(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0. \end{cases} \tag{10.2.1}$$

(b) Ramp with saturation

$$f(x) = \begin{cases} 1, & \text{if } x \geq 1, \\ x, & \text{if } |x| < 1, \\ -1, & \text{if } x \leq 1. \end{cases} \tag{10.2.2}$$

(c) Sigmoid function

$$f(x) = \begin{cases} = \dfrac{1-e^{-x}}{1+e^{-x}}, & \text{for } -1 \le f(x) \le 1, \\ = \dfrac{1}{1+e^{-x}}, & \text{for } 0 \le f(x) \le 1. \end{cases} \quad (10.2.3)$$

The sigmoid function is a differentiable nonlinearity and hence is suitable for continously varying signals.

Neural net structure

(a) Input, output and hidden nodes

Inputs to the input nodes and outputs from the output nodes are directly accessible from the external environment. The external inputs to the network are usually not weighted; all interconnections within the network are weighted. The signals sent out from output nodes can be directly read or measured. The hidden nodes are not directly accessible from the external environment; all input and output connections of these nodes are with nodes within the network only.

In usual configuration, the input nodes constitute the input layer, and the output nodes constitute the output layer of the network. The hidden nodes may belong to one or more hidden layers within the network, which are not directly accessible by the inputs or the outputs.

(b) Architectures

Two basic architectures for neural net are the feedforward and the feedback architectures. Networks may also be designed combining the features of both.

Feedforward architecture
A feedforward neural network has a multilayered structure. The signals flow between the nodes only in the forward direction, i.e. towards the output end (Fig.10.2.2); nodes of a layer can have inputs from nodes of any of the earlier layers. The weights on the nodal interconnections as well as the thresholds are adaptively adjusted to optimize the performance of the network. The popularly used multilayer perceptron has a feedforward architecture.

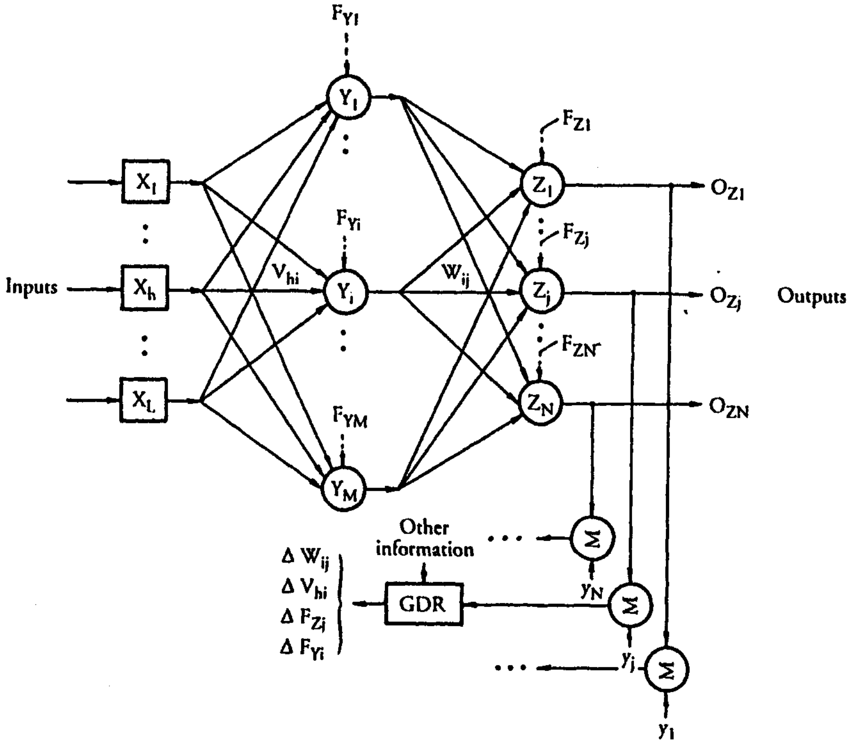


Figure 10.2.2 A three layer perceptron or feedforward network, with schematic arrangement for learning through backpropagation.

Feedback architecture

In the feedback architecture, the output from a node can flow in the forward direction (i.e. to nodes towards the output), or in the reverse direction, or may be fedback as input to the same node itself (see Fig.10.2.3). Such networks are also called *recurrent networks.*

(c) Network adaptation or learning

In neural networks, usually the nodal characteristics remain unchanged. The adjustment of the weights on the nodal interconnections and the thresholds is usually referred to as *training* or *learning* of the network; this is analogous to the estimation of parameters in an identification problem. The learning may be supervised or unsupervised. *Unsupervised learning* is based on maximization of some predefined

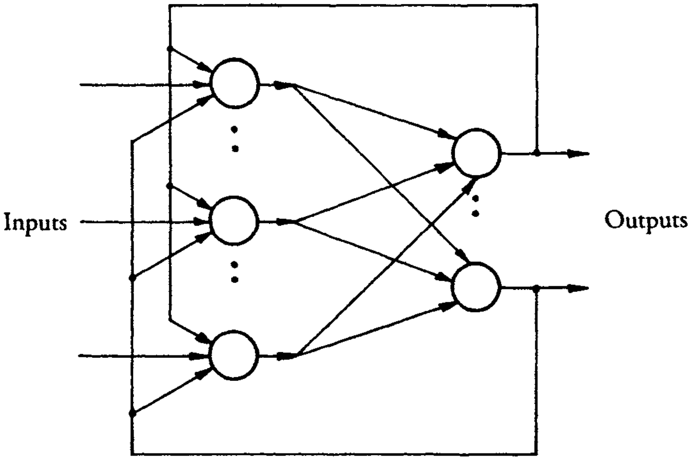


Figure 10.2.3 A two-layer feedback neural network.

function or criterion. *Supervised learning* expects operator intervention. It requires a training data set, comprising a set of input data and a corresponding set of data for the desired outputs; the learning is based on the minimization of the error between the computed and the desired outputs.

The basic error correction rule concerns adjustment of weights on the interconnections in proportion to the error between the computed outputs and the desired values of each node in the output layer. The *gradient descent rule* refers to adjustments of the weights such that the cumulative mean square error over the training set is minimized. The *generalized delta rule* or the *backpropagation (learning) algorithm* is a multilayer learning algorithm based on the gradient descent approach, when the nodal nonlinearity is differentiable like the sigmoidal nonlinearity.

Remarks

(a) The neural net approach presumes knowledge as being built into the nodal interconnections rather than into the outputs from nodes, which may or may not be observable.

(b) The interconnections ascribe robustness to the neural architecture which will work even if some nodes fail.

(c) The ability of a multilayer network in mapping nonlinear relationships comes from the nonlinearities within the nodes. If the nodes had no nonlinearity, there would always be a single layer network which could be functionally equivalent to a multilayer network.

10.3 MULTILAYER PERCEPTRON AND BACKPROPAGATION ALGORITHM

A *multilayer perceptron* is a multilayer feedforward neural network having one input layer, one or more hidden layers and one output layer.

It is found that a three-layer perceptron with the backpropagation learning algorithm can model a wide range of nonlinear relationships to a reasonable degree of accuracy. The ideal structure of the hidden layer and the interconnections are still a subject of research; some results related to the optimization of the size of the feedforward networks are presented in Sec.10.4.

The generalized delta rule (GDR) or the backpropagation algorithm is due to Rumelhart, Hinton and Williams (1986) and is summarized below; detailed derivation follows in Appendix 10.

10.3.1 Backpropagation Learning

Consider a three layer perceptron (Fig.10.2.2) with

(a)	L nodes in the input layer X,	($h = 1$ to L),
(b)	M nodes in the hidden layer Y,	($i = 1$ to M),
(c)	N nodes in the output layer Z,	($j = 1$ to N).

Refer to the desired or test outputs of the network as y_j, and the output of say the i-th node of layer Y as O_{Yi}. The objective is to train the algorithm through adaptation of

(i) the weights W_{ij} and V_{hi} on nodal interconnections between Y and Z layers and between X and Y layers respectively, and

(ii) the threshold values F_{zj} and F_{Yi} for the nodes of Z and Y layers respectively.

The backpropagation learning proceeds as follows:

(1) Initialization:
Assign random values (say, between 0 and 1) to all the weights W_{ij} and V_{hi} and the threshold levels F_{zj} and F_{Yi}.

(2) Output computation:
Read a new set of input data to the network, O_{Xh}, and compute

(i) $$O_{Yi} = f\left(\sum_{h=1}^{L} V_{hi}O_{Xh} + F_{Yi}\right), \qquad (10.3.1)$$

(ii) $$O_{zj} = f\left(\sum_{i=1}^{M} W_{ij}O_{Yi} + F_{zj}\right), \qquad (10.3.2)$$

where $f(.)$ is sigmoidal nonlinearity (10.2.3) between (0,1).

(3) Adaptation of weights:
Read a new set of data for the desired outputs y_j, and compute

(i) $$\Delta W_{ij} = \alpha O_{Yi}D_{zj}, \qquad 0<\alpha<1, \qquad (10.3.3)$$

where

$$D_{zj} = (y_j - O_{zj})O_{zj}(1-O_{zj}),$$

and

$$\Delta W_{ij} = W_{ij}(\text{new value}) - W_{ij}(\text{last value}).$$

(ii) $$\Delta V_{hi} = \beta O_{Xh}D_{Yi}, \qquad 0<\beta<1, \qquad (10.3.4)$$

where

$$D_{Yi} = \left(\sum_{j=1}^{N} W_{ij}D_{zj}\right)O_{Yi}(1-O_{Yi}),$$

(4) Adaptation of thresholds:

$$\Delta F_{zj} = \alpha D_{zj},$$

$$\Delta F_{Yi} = \beta D_{Yi}.$$

(5) Iteration:
Repeat by going to Step (2) and iterate to desired convergence of the computed outputs O_{zj} to the test outputs y_j, which completes the neural network model.

Remarks

(a) The positive constants α and β determine the learning rates which are heuristically chosen to be less than 1. In the examples of this chapter, the values used are $\alpha=\beta=0.4$.

(b) For improved convergence of the algorithm and smoother weight changes, often an additional momentum term is used in (10.3.3) and (10.3.4), which basically acts like a low-pass filter (discussed in Appendix 14A). For example, at k-th iteration,

$$\Delta W_{ij}(k) = \frac{(1-\gamma)\alpha O_{yi} D_{zi}}{(1-\gamma q^{-1})}, \qquad 0<\gamma<1.$$

10.3.2 Application Example

The modelling of a nearly periodic process is discussed here. More complex examples are given in Secs.10.4 and 10.5.

A three layer neural network is used. Each of the hidden and the output nodes comprises a summer with threshold followed by sigmoidal nonlinearity (between 0-1). The network is trained using the backpropagation algorithm. The inputs to the network and the reference output are normalized to lie within 0.3 and 0.7 (that is for the data to remain approximately within the linear region of the sigmoidal nonlinearity).

Example 10.3.2 Modelling and prediction of Trans-Atlantic Airline passenger series

This series contains monthly air-traffic data over 12 years (see Appendix 7A.2); the data show yearly periodicity.

There can be different ways of modelling the airline traffic series, depending on the objective. If the model has to produce 12 step ahead prediction, a typical representation can be

$$y(k) = f(y(k-12), \ y(k-24)).$$

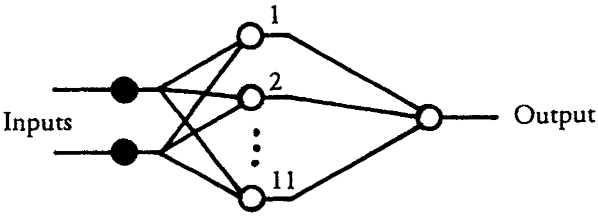


Figure 10.3.1 A 2-11-1 feedforward network modelling the Trans-Atlantic Airline traffic series.
o, a node with nonlinearity; •, a unity gain node.

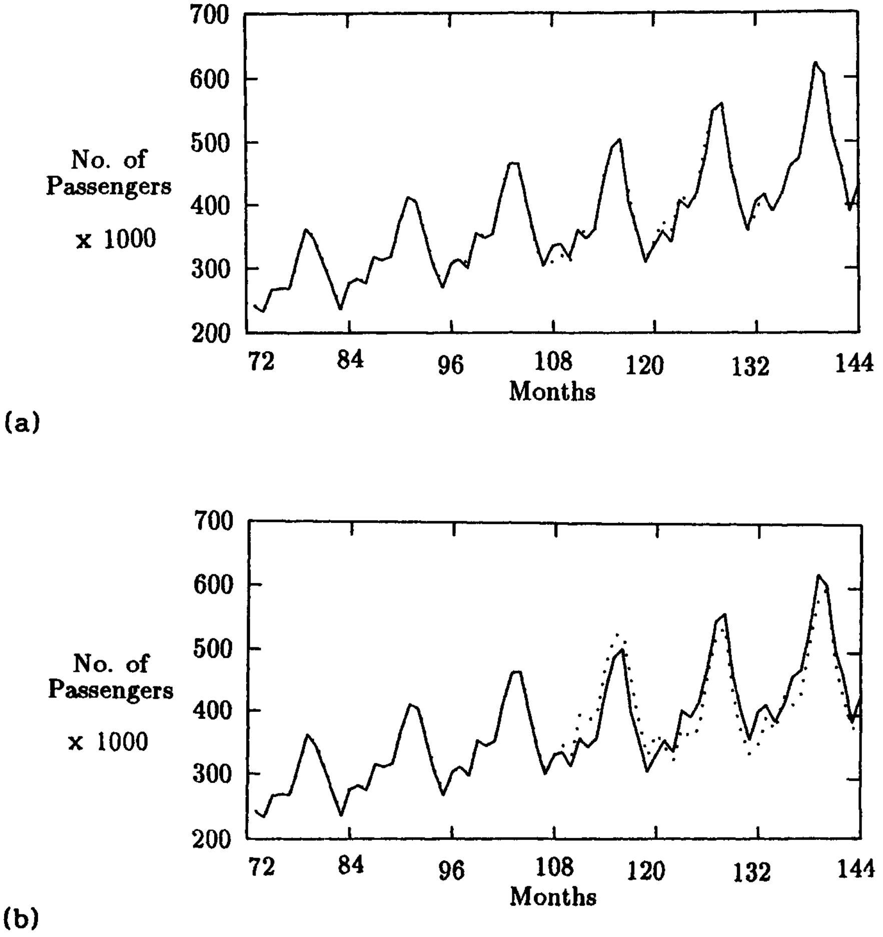


Figure 10.3.2 (a) Training of the Air traffic series. (b) One period ahead prediction for 10th to 12th years.

This process can be modelled by a three layer neural network (Fig.10.3.1) with 2 input nodes and 1 output node with adequate number of hidden layer nodes (as discussed later). Let the data for the 7th and 8th year be used as input and the data for the 9th year be used as the output. The input data constitute a 2x12 matrix, and the output data are a 12 element vector. Here the network has to learn 12 patterns, so 11 hidden nodes are used. The internodal links and the thresholds are initialized with random values between -1 and

1. The network is trained over 30,000 epochs (an epoch being a training event corresponding to a single pass over the entire training set). The trained network is used to predict the air-traffic for the 10th year.

Again the network is trained with the data for the 8th, 9th and 10th years to predict the same for the 11th year. Similarly the traffic for the 12th year is also predicted. The training results over the periods 8th to 11th are shown in Fig.10.3.2(a), and the prediction results over 10th to 12th periods are shown in Fig.10.3.2(b).

Remarks
The number of nodes required in the hidden layer depends on the inputs of the network as well as the number of patterns to be learnt (which is the same as the epoch length in the present context). For N number of input nodes usually 2N+1 number of hidden nodes are used; again for learning M different patterns, maximum M-1 number of hidden nodes are required, although the optimum number of hidden layer nodes required can be much less as discussed in the following two sections.

10.4 DESIGN OF OPTIMUM NETWORKS USING SVD AND SUBSET SELECTION

In any method of modelling, overparameterization or redundancy in the structure is undesirable. A neural network will be overparameterized if the number of links is excessive. In such cases, if the training set of data are not noise-free, the network will tend to learn the information along with noise in the data leading to poor validation results.

For an optimum design, the neural network should have the optimum number of inputs, and the optimum number of links and nodes within the network. This section addresses two basic questions, namely (i) *which* of the *candidate inputs* to the network constitute the optimal set of inputs, and (ii) at the hidden layer(s), *which nodes and links* between the post-hidden layer stage and the subsequent stage(s) are essential for the design of the network with optimum size. The singular value decomposition (SVD) and subset selection (Sec.3.6.2) based on QR with column pivoting (QRcp) factorization are used for the design of feedforward networks of optimum size.

10.4.1 Determination of Optimum Architecture

Input nodes

The basic problem is to determine the optimum number of inputs that should be considered. This involves subset selection from the data sets of all candidate inputs for the selection of the input variables which contain significant information. SVD followed by QRcp factorization can be used for subset selection as discussed in Sec.3.6.2.

Optimum number of hidden nodes and links

At any post hidden layer stage, the links that are connected with any subsequent layer act as inputs to those layers. It is desirable to eliminate those links which are redundant, or which carry relatively insignificant information.

Two possible network architectures are considered:

(a) *Homogeneous networks*: Here all the input nodes are connected with all the hidden nodes only (e.g., see Fig. 10.4.1a). This is the most popular architecture.

(b) *Nonhomogeneous networks*: This covers all the designs excluding the homogeneous networks. In these architectures, all possible combinations of inputs are fed to the hidden layer nodes; the direct links bypassing the hidden layer nodes are also permitted (e.g., Fig.10.4.2a).

In the first case SVD can be used to determine how many hidden layer nodes should be used; in the second case subset selection can be performed to determine which links between the hidden layer stage and the subsequent stage can be eliminated without any appreciable degradation of performance.

The problem of modelling a multi-input single-output process using a 3-layer neural network is detailed below.

Design procedure

Let there be n candidate inputs (i.e. the input vector is n-dimensional) and one output; suppose there are m sets of input and output data available.

(1) First a candidate network is considered, which may be exhaustive or overparameterized but not underparameterized. This network is iterated to *crude convergence* (explained later).

(2) Suppose the number of links between the hidden layer

and the subsequent layer (i.e. the number of links to the output layer in the present case) is r; the r links are referred to as the *pseudo-outputs* of the hidden layer. Here r includes the number of hidden layer nodes as well as the number of direct links bypassing the hidden layer (that is the direct links between the earlier layer(s) and layer(s) following the concerned hidden layer). For each of the m input data sets to the network, compute the r pseudo-outputs of the hidden layer; an mxr matrix **B** is formed corresponding to the m sets of input data.
(3) SVD is performed on **B**: $\mathbf{B} = \mathbf{U}_B\mathbf{S}_B\mathbf{V}_B^T$. The number of dominant singular values in $\mathbf{S}_B$ (say g, $g \le r$), indicating the rank or appoximate-rank of **B**, will indicate the *number* of links that should be retained.

Remark: For homogeneous networks, the selection stops here. For nonhomogeneous networks, the selection procedure continues as follows.

(4) QRcp factorization is performed on gxr matrix $\bar{\mathbf{V}}_B^T$ for subset selection, and the *specific* g *of* r *links* between the hidden layer and the subsequent layer to be retained are identified; here $\bar{\mathbf{V}}_B$ contains the first g columns of $\mathbf{V}_B$.

The reduced-size network is reinitialized and retrained to the desired convergence.

Justification

The objective is to eliminate collinearity or near-collinearity between the different variables (or links carrying information), which is done using the numerically robust approach of SVD and QRcp factorization. The selection can be unique if the distribution of the singular values shows a large drop (as explained in Sec.3.6.2). Following (7.6.6), the squared sum of the eliminated singular values quantifies the part of the available energy in the data, which is rejected in the subset selection procedure.

Remarks
(1) The state of 'crude convergence' mentioned in step (i) above is not uniquely defined. The selection is expected to be meaningful, when the distribution of the singular values shows a relatively large drop. According to experimental verifications, the selection may be possible quite early in

the training.

(2) If there are more than one hidden layers, the selection procedure starts with the first hidden layer, and the whole exercise is repeated sequentially for each subsequent hidden layer towards the output.

(3) If only one set of input data is available for the training of the network (i.e. m=1), **B** at step (2) above can be formed from the outputs at the post hidden layer stage for m (>1) consecutive iterations. Step (3) and step (4) can follow subsequently the same way as above.

(4) There are alternative approaches for the reduction of the size of the network. For example, Karhunen-Loève transformation (KLT) and principal component analysis based methods have been proposed; both these are eigenvalue based, which are numerically less robust than the singular value based methods.

10.4.2 Modelling and Prediction of Mackey-Glass Series

The Mackey-Glass equation (Mackey and Glass, 1977), which models the nonlinear oscillations occurring in physiological processes, has been discussed in Sec.8.3.1 and Example 8.3.2. Consider a discrete-time representation of the Mackey-Glass (MG) equation given by

$$x(k+1) - x(k) = \frac{0.2x(k-\tau)}{1 + x^{10}(k-\tau)} - 0.1x(k), \qquad (10.4.1)$$

$$\tau = 17,$$

which generates a quasiperiodic series as shown in Example 8.3.2. The objective is to model the MG series and to produce multistep ahead predictions.

Here the series $\{x(k)\}$ can be expressed as

$$x(k+p) = f(x(k), x(k-\tau), x(k-2\tau), \ldots, x(k-(N-1)\tau)), \qquad (10.4.2)$$

where p is the prediction time which can be chosen depending on short-term or long-term prediction; it has been shown (Lapedes and Farber, 1987) that N can be typically between 4 and 8. In the present case N=6 is chosen; another reason for this choice is that the series (10.4.1) has a pseudo period length varying between 95-102, which is almost covered by the input data sets in (10.4.2).

Three-layer feedforward networks are used with sigmoidal nonlinearity (between 0 and 1) and the backpropagation algorithm is used for training. A six input network is considered where x(k), x(k-τ),..., x(k-5τ), with $\tau = 17$, are used as the inputs and x(k+p) is used as the output. For all exercises 300x6 data set is used for training, and the subsequent 200x6 data set is used for validation test through 6-step ahead prediction with p=6. The input data set may be represented by the matrix

$$\mathbf{A} = \begin{bmatrix} x(k-6) & x(k-23) & x(k-40) & x(k-57) & x(k-74) & x(k-91) \\ x(k-7) & x(k-24) & x(k-41) & x(k-58) & x(k-75) & x(k-92) \\ x(k-8) & x(k-25) & x(k-42) & x(k-59) & x(k-76) & x(k-93) \\ & & \vdots & & & \end{bmatrix};$$

the corresponding 300x1 output data set is given by

$$\mathbf{y} = [x(k) \quad x(k-1) \quad x(k-2) \ \ldots \ ]^T.$$

The modelling exercises used follow.

Exercise 1 Modelling of the Mackey - Glass series with a homogeneous neural network

(a) Design and selection:
A network with 6 inputs, 11 hidden nodes and 1 output is considered (which is referred to as a 6-11-1 network) (see Fig.10.4.1(a)). Throughout the training, SVD is performed on the 99x11 matrix **B**, which is a subset of the available 300x11 matrix at the post hidden layer stage to determine the optimum number of hidden nodes (the size of **B** is not a limitation). The distribution of the singular values is shown in Fig.10.4.2; these results lead to the deduction that 3 to 4 singular values are relatively dominant. So a reduced 6-3-1 network (Fig.10.4.1b) is considered. Further subset selection is not necessary since inputs to the nodes are similar.

(b) Training and validation:
Both 6-11-1 and the reduced 6-3-1 networks are trained to convergence and the validation is tested. The results (Figs.10.4.3, 10.4.4, and Table 10.4.3) show that the convergence rate and the performance of the reduced network is comparably good. In Fig.10.4.3 both fit for the training data set and 6-step ahead prediction results are presented.

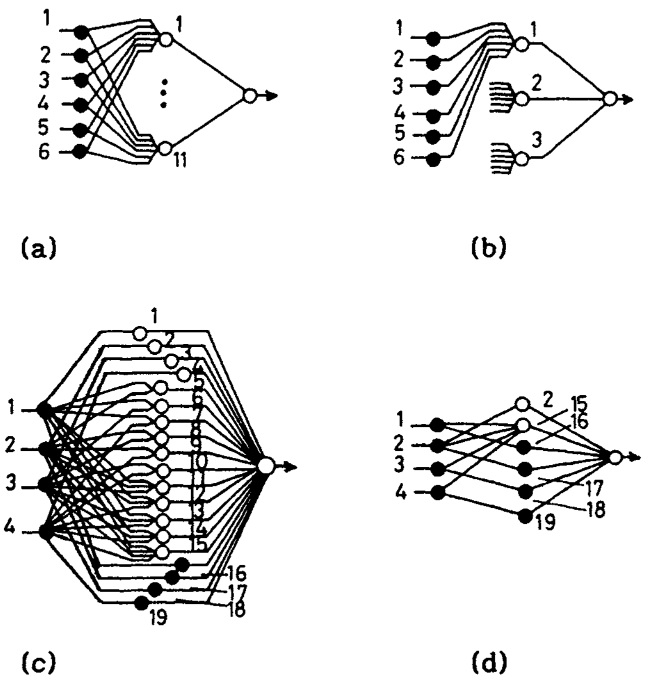


Figure 10.4.1 Neural networks modelling the MG series: (a) 6-11-1 network, and (b) the reduced 6-3-1 network; (c) 4-19-1 network, and (d) the reduced 4-6-1 network. o, a node with nonlinearity; ●, a unity gain node.

Exercise 2 Modelling of the Mackey-Glass series with an optimal network

(a) Input selection:
Subset selection is performed on different blocks of data from the 300x6 input data set and the optimal set of 4 inputs in (10.4.2) are determined (Table 10.4.1).

(b) Design and selection:
An exhaustive 4-19-1 network is considered as shown in Fig. 10.4.1(c). The network is trained with the selected 300x4 data set. During the training, corresponding to 90 input data sets (out of 300), the respective magnitudes of the variables at the links at the post hidden layer stage are used to form the 90x19 matrix **B**. At some selected points

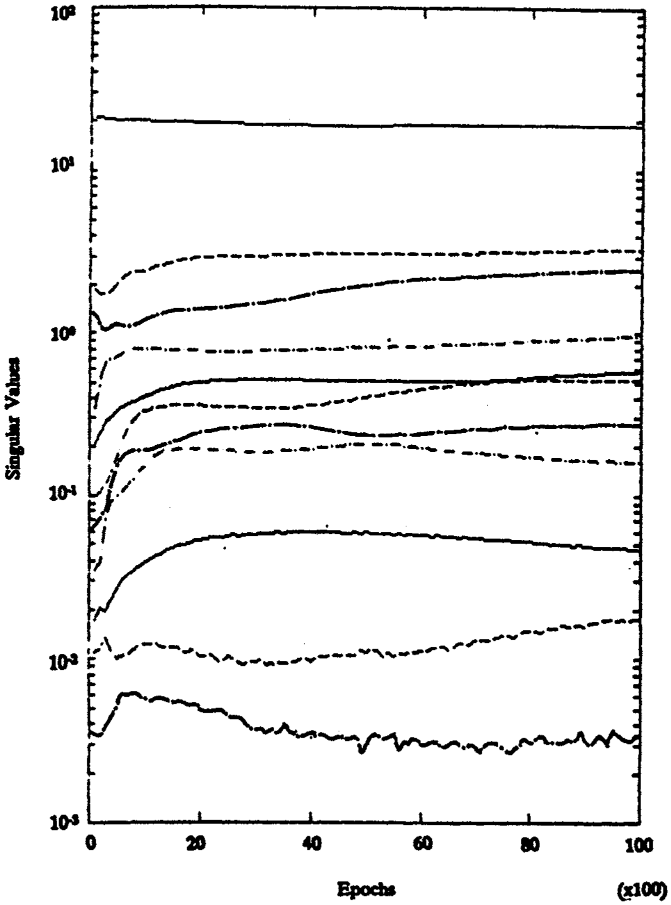


Figure 10.4.2 Singular value distribution of **B** at post hidden layer stage during the training of 6-11-1 network.

subset selection is performed on **B**. The results (Table 10.4.2) show that out of the 19 links at the post-hidden-layer stage, only 6 specific links carry the major part of the information, leading to the reduced 4-6-1 network shown in Fig.10.4.1(d).

(c) Training and validation:
Both the 4-19-1 and the reduced 4-6-1 network are trained to convergence and the validation is tested; the results (Figs.10.4.5, and Table 10.4.3) show that the performance of the reduced network is equally good.

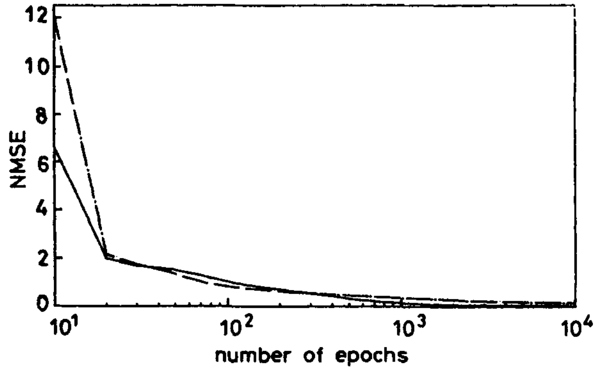


Figure 10.4.3 NMSE during training of MG series. ——— 6-11-1 network, —·—· 6-3-1 network.

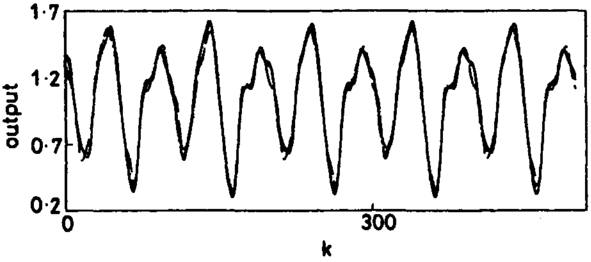


Figure 10.4.4 MG series through 6-11-1 and 6-3-1 networks. ——— original data, ——- 6-11-1, —— – 6-3-1 networks.

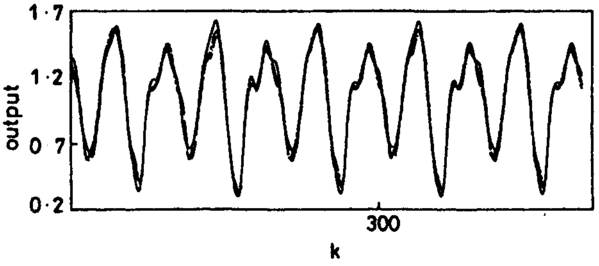


Figure 10.4.5 MG series through 4-19-1 and 4-6-1 networks. ——— original data, ——- 4-19-1, —— – 4-6-1 networks.

Table10.4.1 Selection of optimal input set modelling the MG series (out of 6 candidate inputs)

Data blocks	Singular values	Selected input variables
(1 to 90) x 6	12.8, 3.4, 3.2, 1.3 0.880, 0.701	$x(k-2\tau)$, $x(k-5\tau)$, $x(k)$, $x(k-4\tau)$
(200 to 290) x 6	12.8, 3.4, 3.1, 1.3 0.864, 0.805	$x(k-2\tau)$, $x(k-5\tau)$, $x(k)$, $x(k-4\tau)$

Table 10.4.2 Selection of optimal links at hidden layer of 4-19-1 network (Fig.10.4.1(c))

Epochs	Singular Values of 90x19 matrix **B**	Selected links
10	28.1, 1.3, 0.9, 0.7, 0.6, 0.2, 0.115, . . . , 0.003	16,18,17,19,15,2
1000	22.2, 3.3, 2.6, 1.3, 0.6, 0.4, 0.126, . . .0.002	15,18,16,17,19,2

Table 10.4.3 Normalised RMSE (√(MSE/variance)) for 6-step ahead prediction over the data set: 301-500

Network:	6-11-1	6-3-1	4-19-1	4-6-1
NRMSE	0.137	0.092	0.139	0.152

10.4.3 Modelling of Chemical Oxygen Demand (COD) in Osaka Bay

The Mackey-Glass series modelled in Sec.10.4.3, was a case where the data were noise-free. Here is an example of a real-life input-output process with noisy data.

The COD process in the Osaka Bay (Appendix 9) is a three input, one output process. In Sec.9.4, this process has been modelled using GMDH; here neural network models are considered. Out of the available data, the first 60 sets are used for modelling the network and the next 23 sets are used for validation tests.

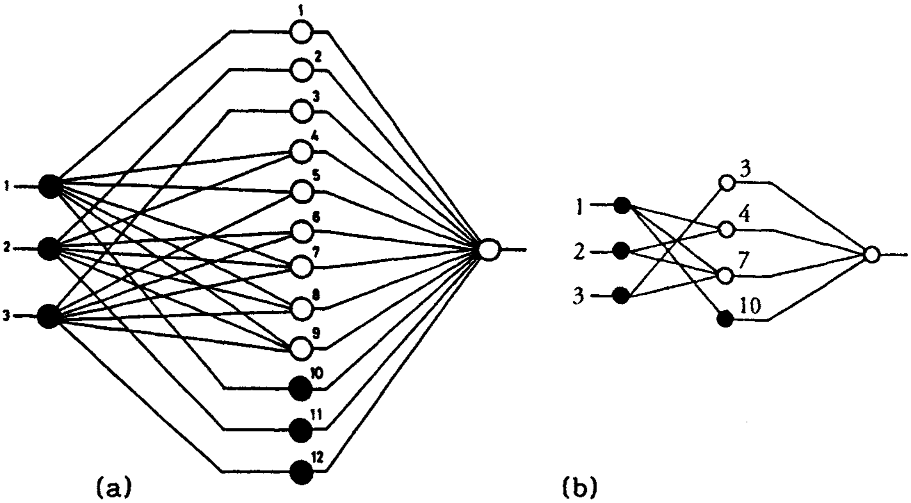


Figure 10.4.6 Nonhomogeneous neural networks modelling the COD process:
(a) 3-12-1 network, and (b) the optimised 3-4-1 network.

Exercise 1 Modelling of the COD process using a homogeneous neural network

A homogeneous network with 3 input nodes, 6 hidden layer nodes and 1 output node is considered (the network is structurally similar to Fig.10.4.1(a)). The training is performed for 60 input data sets which are fed to the network sequentially. The matrix **B** formed from the data at the post hidden layer stage during the course of training is SV-decomposed. The distribution of the singular values (presented in Table 10.4.4) shows only four singular values being relatively dominant; so 4 hidden nodes should be adequate for modelling.

Both the 3-6-1 network and the reduced 3-4-1 networks are trained to convergence, and the validation is tested. The output estimation error for the input data over the validation period are presented in Table 10.4.6.

Exercise 2 Modelling the COD process with an optimised structure.

An 3-12-1 network with exhaustive choice of connections between the input and output layers is considered

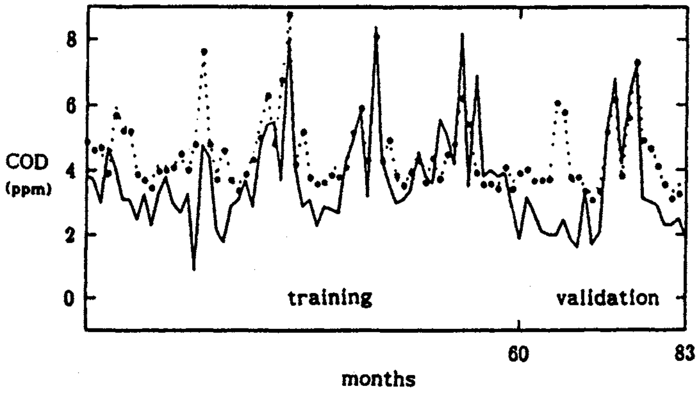


Figure 10.4.7 Modelling and validation of COD (output) ——— data, ... 3-12-1 network, ... 3-4-1 network.

(Fig.10.4.6(a)). At different stages of training SVD followed by subset selection is performed on matrix **B** at the post hidden-layer stage. It is found that 4 out of the 12 links carry significant information (see Table 10.4.5), which are links 4,7,3, and 10; this led to a 3-4-1 network (Fig. 10.4.6(b)).

Both 3-12-1 and 3-4-1 networks are trained to 30,000 epochs; their comparative performances as shown in Fig.10.4.7 and Table 10.4.5 are found to be reasonably close.

Remarks

(1) The selection of the hidden layer links for the optimised structure may not be unique over the learning regime of the network, as in Exercise-2 on the COD process (see Table 10.4.5). In the present case, the validation performance for the candidate networks (4,7,9,10) and (4,7,3,10) however show close performance. Thus it is expected that in a real-life situation, the presented method for the optimised structure will lead to a solution with significantly reduced size, although the closeness to optimality is difficult to quantify, except through tests like validation performance etc.

(b) Both in cases of the MG series modelling (Sec.10.4.2) and the COD process modelling, the SVD and subset selection based method of producing reduced-size networks is found to work. As expected the reduced-size networks show comparable

Table 10.4.4 Selection of optimum number of hidden nodes for 3-6-1 homogenous network modelling the COD process

Epochs	Singular values	No. of nodes
20	7.58, 0.56, 0.26, 0.17, 0.007, 0.004	4
200	8.43, 0.79, 0.41, 0.22, 0.012, 0.005	4
2000	11.72, 1.86, 0.59, 0.14, 0.044, 0.005	4

Table 10.4.5 Selection of optimal links at hidden layer of 4-19-1 network modelling MG series

Epochs	No. of nodes selected	Singular values	Selected links
200	4	16.31, 1.15, 0.54, 0.29, 0.019,..., 0.002	4,7,9,10
2000	4	9.60, 1.14, 0.54, 0.32, 0.04,...., 0.002	4,7,3,10

Table 10.4.6 Normalized RMSE for estimation of COD over the validation data sets

Network:	3-6-1	3-4-1	3-12-1	3-4-1
NRMSE	0.1696	0.1653	0.139	0.123

or better performance than the oversized networks. In the COD process case, the data being noisy, the reduced nonhomogeneous network produces minimum error on validation test (see Table 10.4.6).

10.5 MODELLING NETWORKS WITH ORTHOGONALIZED DATA

The neural networks designed in the last section used measured input and output data in time (or spatial) domain. This section studies neural network models for nearly periodic series using orthogonalized data sets. The method is also applicable for quasiperiodic series configured as periodic series, as discussed in Sec.11.2.1.

10.5.1 Modelling Principle and Perspective

Periodic data series can be optimally compressed using SVD; this feature is used in present modelling procedures.

Orthogonalization

An mxn matrix $\mathbf{A} = \mathbf{USV}^T$ can be orthogonalized as $\mathbf{Z} = \mathbf{AV} = \mathbf{US}$, where $\mathbf{Z} = [\mathbf{z}_1, .., \mathbf{z}_i, .., \mathbf{z}_n]$; $\mathbf{z}_i$ are m-orthonormal vectors: for example, $\mathbf{z}_1 = [z_{1,1} \ldots z_{i,1} \ldots z_{m,1}]^T$. If Rank($\mathbf{A}$) = p,

$$\mathbf{A} = \sum_{i=1}^{p} \mathbf{u}_i \sigma_i \mathbf{v}_i^T = \sum_{i=1}^{p} \mathbf{z}_i \mathbf{v}_i^T. \qquad (10.5.1)$$

If p=1, the i-th row of $\mathbf{A}$ is given by $z_{i,1}\mathbf{v}_1^T$.

Arrangements and analysis of the data

The characterization of a nearly periodic series through singular value decomposition has been discussed in Sec. 7.7.1; some of the main points are restated here.

The time series data can be arranged into a matrix $\mathbf{A}$ such that the consecutive periods are aligned into consecutive rows. The degree of periodicity will be reflected in the singular values of $\mathbf{A}$. The ratio, s_1/s_2, will be ∞ for a strictly periodic series; as the series deviates from perfect periodicity this ratio decreases.

Modelling and prediction

When s_1/s_2 is large, $\mathbf{A} = \mathbf{USV}^T \propto \mathbf{u}_1 s_1 \mathbf{v}_1^T = \mathbf{z}_1 \mathbf{v}_1^T$, where $\mathbf{v}_1^T$ will be indicative of the periodic pattern, and the elements of $\mathbf{z}_1$ will be the scalar weights associated with the respective rows of $\mathbf{A}$. For the sake of simplicity let the elements of $\mathbf{z}_1$ be expressed as

$$\mathbf{z}_1 = [z_1, z_2, \ldots, z_m]^T.$$

The series of elements $\{z_1, z_2, \ldots, z_m\}$ are modelled using a neural network. The modelling scheme assumes that $\mathbf{v}_1^T$, the first right singular vector, remains almost unchanged if an additional row is appended to $\mathbf{A}$.

So, one period ahead prediction of the time series represented by $\mathbf{A}$ will be given by $\hat{z}_{m+1}\mathbf{v}_1^T$, where $\hat{z}_{m+1}$ is obtained from the neural network.

The modelling scheme is presented in Fig.10.5.1(a). The orthogonalized data are arranged for modelling as follows.

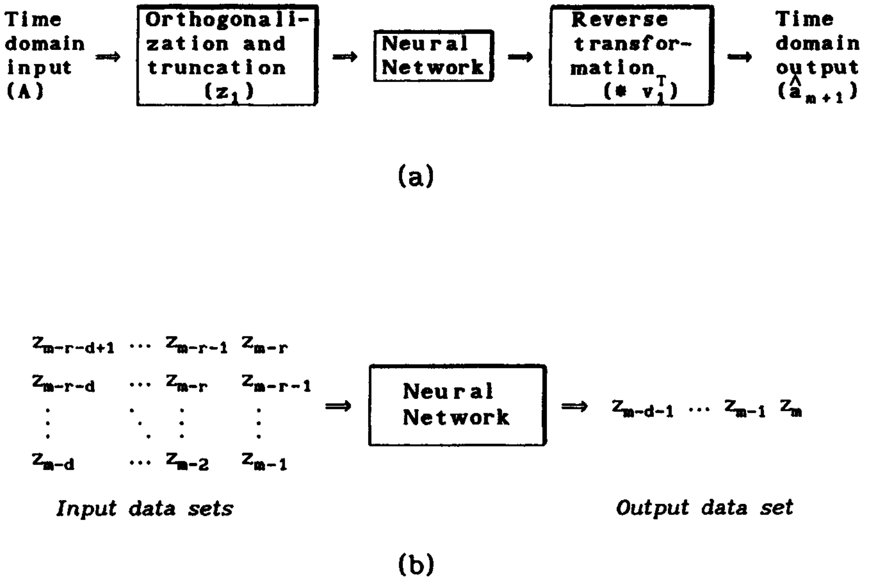

Time domain input (A)
Orthogonalization and truncation (z_1)
Neural Network
Reverse transformation ($* \, v_1^T$)
Time domain output ($\hat{a}_{m+1}$)
(a)
$z_{m-r-d+1}$ ⋯ z_{m-r-1} z_{m-r}
z_{m-r-d} ⋯ z_{m-r} z_{m-r-1}
z_{m-d} ⋯ z_{m-2} z_{m-1}
Neural Network
z_{m-d-1} ⋯ z_{m-1} z_m
Input data sets
Output data set
(b)

Figure 10.5.1 (a) Periodic series modelling scheme with the neural networks operating with orthogonalized data. (b) The input and output data configurations.

Consider a neural network with r inputs and 1 output; as shown in Fig.10.5.1(b), let the data be configured for an epoch length of d (≥ 1), where $(r+d) \leq m$. Thus the input and the output data sets are suitably arranged from the data sets $\{z_{m-1}, z_{m-2}, \ldots, z_{m-r-d+1}\}$ and $\{z_m, z_{m-1}, \ldots, z_{m-d+1}\}$ respectively, for the training of the network. The trained network is subsequently used with the corresponding input set configured from the data $\{z_m, z_{m-1}, \ldots, z_{m-r-d}\}$ to produce the predicted output vector $\{\hat{z}_{m+1}, \hat{z}_m, \ldots, \hat{z}_{m-d}\}$. The periodic prediction is given by $\hat{z}_{m+1} v_1^T$.

The neural network may be homogeneous or nonhomogeneous, and may be designed using the concepts discussed in the earlier sections.

Summary

(1) The m consecutive periods (each of length n) of the series are aligned into consecutive rows of an mxn matrix **A**.

(2) Compute $\mathbf{A} = \mathbf{USV}^T = \mathbf{ZV}^T$; if $s_1 >> s_2$, $\mathbf{A} \simeq \mathbf{z}_1\mathbf{v}_1^T$.
(3) The series formed by the elements of $\mathbf{z}_1$ are modelled using (typically) a r-1-1 feedforward network ($r \leq m-1$).
(4) The trained network is used to produce the prediction $\hat{z}_{m+1}$.
(5) The one period ahead prediction is produced as $\hat{z}_{m+1}\mathbf{v}_1^T$.

Remarks
(1) If more than *one* (SV-)decomposition components are dominant in (10.5.1), separate neural networks will be required for modelling each (see Sec.7.8.1).
(2) The main strengths of the present method are (a) the modelling in terms of the SV-decomposition components which carry compressed information by virtue of the transformation, and as a result the network size is substantially reduced, and (b) the *nonlinear* modelling of the elements of $\mathbf{z}_1$ through the neural network, which makes the present method different from the one presented in Sec.7.8.1.

10.5.2 Modelling of the Indian Rainfall Series

A series of spatially coherent rainfall over the North-western and Central parts of India is considered (Appendix 7F). The data for 40 years (from 1940) are used for modelling. The data are monthly with yearly periodicity. The series shows poor degree of repeatibility (see Fig.2.2.2) with s_1/s_2 of the order of 6.

The data are arranged as shown in Fig.10.5.1(b); it is found that an appropriate choice can be d=34, and r=6. A 3 layer network with sigmoidal nonlinearity at the hidden and output nodes is considered, and the backpropagation algorithm is used for training. A 6-3-1 homogeneous network is used, which is trained through 5000 epochs. The prediction result for 3 consecutive years is shown in Fig.10.5.2.

Remark
In the present context, the poor repeatibility between the periods may be due either to the variation of the periodic pattern or to the variations in the scaling factors between the patterns, or both. The present approach assumes the pattern remains almost unchanged at $\mathbf{v}_1^T$, so the variation in the scaling factor (given by the elements of $\mathbf{z}_1$) is modelled.

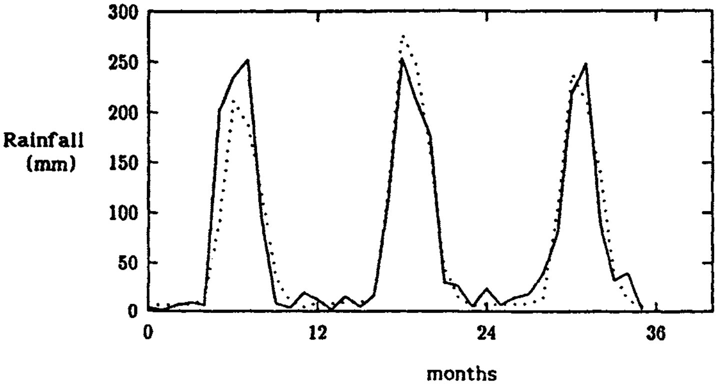


Figure 10.5.2 1 to 12 month ahead prediction of rainfall over the period 1981-1983;
——— actual data, ····· predicted values.

10.6 ASSESSMENT OF CONVERGENCE USING SVD

The convergence of a neural network during training is usually assessed in terms of the output error. If an mxn input data set is used for training, the output errors for m different sets of input has to be studied. SVD offers an alternative approach to convergence assessment through the assessment of rank-oneness of the output matrix over several epochs as follows.

The training through one m-long epoch implies m number of network-weight updations, which will produce an m-output vector; let the corresponding reference output vector be y_R. g epochs will produce an mxg output matrix Y. At *ideally true* convergence, all the columns of Y will be identical to y_R, and Y will be of unit rank. On the other hand, during the training, before convergence is reached, the weights on the links and thresholds will keep changing, and in such a state, the columns of Y will be different from each other. So the degree of convergence of the network can be expressed in terms of the distance of the output matrix Y from rank-oneness, or in terms of the singular values as follows.

Let SVD of Y be expressed as

$$Y = \sum_{i=1}^{p} u_{Yi} s_{Yi} v_{Yi}^T, \qquad (10.6.1)$$

where p = min(m,g). The ratio of the energy contained in the most dominant component $u_{Y1}s_{Y1}v_{Y1}^T$ and the total reference-output-energy is given by

$$c = s_{Y1}^2/gy_R^Ty_R.$$

Ideally, at convergence c = 1, so the percentage of residual energy at convergence can be expressed as

$$\kappa = 1 - c.$$

Remarks

(1) κ will be insensitive to local minima if g is large enough to encompass such minima.

(2) The NN output and the reference data sets are mean extracted before computing κ to make s_{Y1} insensitive to the mean value for nonzero-mean data.

Example 10.6 Convergence assessment of the 6-3-1 homogeneous network modelling the Mackey-Glass series

This network is shown in Fig.10.4.1(b). At different stages during the training of the network with an epoch length of 209, the output matrix Y is formed with g = 200. The convergence results are shown in Fig.10.6.1. The learning activity shown by the κ profile conforms to that shown by the output error plot.

10.7 CONCLUSIONS

The neural network is an extremely versatile method for modelling and prediction, which is applicable to a large number of problems. In addition, neural networks are amenable to high speed processing through parallel computing, which greatly extends the application prospects for these networks.

A neural network is constructed with weighted interconnections of the basic building blocks called nodes, each of which represents simple and static nonlinearity. One of the strengths of the neural network modelling is the adaptive learning or weight-updating mechanism which is based on an iterative nonlinear optimization technique. The networks are much more flexible, fault-tolerant and powerful than GMDH models discussed in the last chapter.

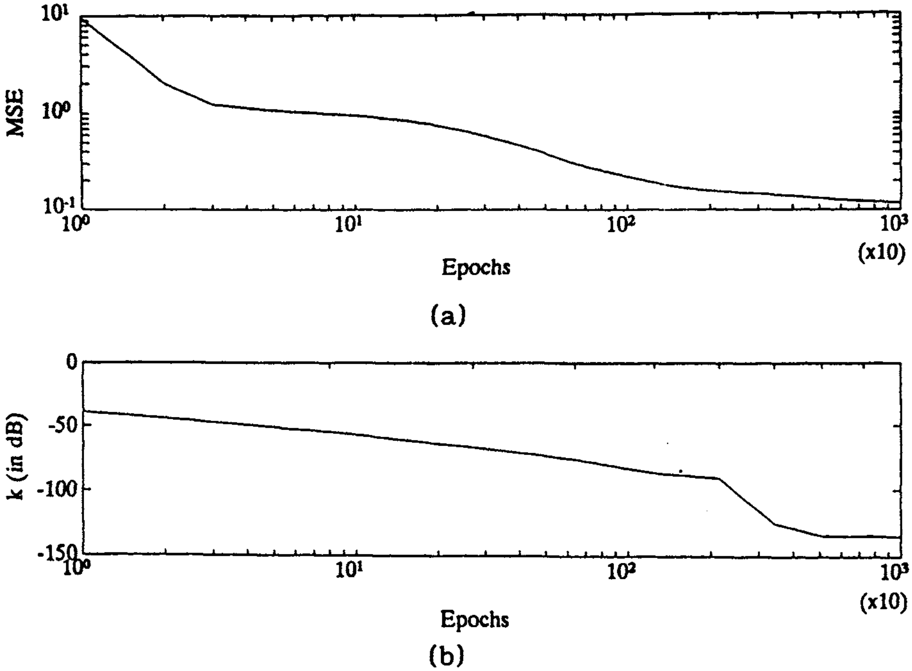


Figure 10.6.1 Assessment of the convergence during training of the 6-3-1 network modelling the MG series: (a) output error approach (b) SVD based approach.

As regards the size of the network, it is important that the network is of optimum size; a oversized network is expected to model the process as well as the noise, whereas the undersized network will not be able to represent the process dynamics faithfully. In this chapter the design of networks with seemingly optimised structure using SVD and QRcp factorization based subset selection has been discussed and demonstrated; the high degree of numerical robustness of these transforms is an important feature. In this connection, it is also to be noted that a conventional homogeneous network may not always be the best architecture for a real-life process.

An SVD based method for the convergence assessment in the training of the network has also been discussed. The convergence over a number of epochs is estimated through the rank-oneness assessment of the output error matrix, which can be more meaningful than the conventional output error method.

REFERENCES

Remarks: Some classic works relating to neural networks are [2,14,20]. The development of the backpropagation algorithm in [15] provided a great thrust to the subject of neural networks. Reviews of neural network models appear in [10,5]; the applications in the control area are reviewed in [4]. Many papers discuss designing with network architectures aiming to be optimal; an information criterion based assessment of optimum size of the network is studied in [1], use of principal components features in [9], use of KLT is discussed in [12], some of the other alternative approaches feature in [16,19]; the hiddden layer sizing aspects are discussed in [17]. The application of SVD and QRcp factorization for optimal design of networks is discussed in [6,7]. System identification and parameter estimation are discussed in Chapter 3. Neural network models of time series appear in [13]. There are many texts on neural networks, e.g., [3,18].

[1] Fogel, D.B. (1991): 'An information criterion for optimal neural network selection', *IEEE Trans Neural Network*, 2(5), 490–497.

[2] Gabor, D., W.P.L. Wilby, and R. Woodcock (1961): 'A universal nonlinear filter predictor and simulator which optimizes itself by a learning process', *IEE Proc.*, Pt.B, **108**, 422–433.

[3] Hertz, J., A. Krogh, and R.G. Palmer (1991): *Introduction to the Theory of Neural Computation*, Addision Wesley, California.

[4] Hunt, K.J., D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop, (1992): 'Neural networks for control systems - a survey', *Automatica*, **28** (6), 1083–1112.

[5] IEEE Proceedings (1991): Special issues on Neural Networks, **78**, Nos. 9–10.

[6] Kanjilal, P.P., P.K. Dey, and D. Banerjee (1993): 'Reduced-size neural networks through singular value decomposition and subset selection,' *Electronics Letters*, 29(17), 1516–1518.

[7] Kanjilal, P.P., and D.N. Banerjee, (1995): 'On the application of orthogonal transformation for the design and analysis of feedforward networks', *IEEE Trans. on Neural Networks*, 6(5), Sept.

[8] Lapedes, A., and R. Farber (1987): 'Nonlinear signal processing using neural networks', Los Alamos National Lab. LA-UR-2662.

[9] Levin, A., T.K. Leen, and J.E. Moody (1994): 'Fast pruning using principal components', in *Advances in Neural Information Processing Systems 6*, Eds. J.D. Cowan, G. Tesauro, and J. Alspector, Morgan Kaufman Publishers, San Fransisco.

[10] Lippmann, R.P. (1987): 'An introduction to computing with neural nets', *IEEE ASSP Magazine*, 4, No.2, 4-22.

[11] Mackey, M.C., and L. Glass, (1977): 'Oscillations and Chaos in Physiological Control Systems', *Science*, 197, No. 4300, 287-289.

[12] Malki, H.A., and A. Moghaddamjoo (1991): 'Using the Karhunen-Loève Transformation in the Back-Propagation Training Algorithm', *IEEE Trans. on Neural Networks*, 2(1), 162-165.

[13] Park, D.C., M. El-Sharkawi, R.J. Marks II, and L. Atlas (1991): 'Electric load forecasting using an artificial neural network', *IEEE Trans. Power Systems*, 6, 442-449.

[14] Rosenblatt, F. (1962): *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington.

[15] Rumelhart, D.E., G.E.Hinton and R.J.Williams (1986): 'Learning internal representations by error propagation', in *Parallel Distributed Processing*, Vol. 1, Ch. 8., D.E. Rumelhart, and J.L. McClelland, Eds., M.I.T. Press, Cambridge, MA.

[16] Sanger, T.D. (1991): 'A tree structured algorithm for reducing computation in networks with separable basis functions', *Neural Computation*, 3, 67-78.

[17] Sartori, M.A., and P.A. Antsaklis (1991): 'A simple method to derive bounds on the size and to train multilayer neural networks', *IEEE Trans. on Neural Networks*, 2(4), 467-471.

[18] Simpson, P.K. (1990): *Artificial Neural System*, Pergamon Press, New York.

[19] Tenorio, M.F., and W. Lee (1990): 'Self-organizing network for optimum supervised learning', *IEEE Trans. on Neural Networks*, 1(1), 100-110.

[20] Widrow, B., and M.E. Hoff (1960): 'Adaptive switching circuits', 1960 IRE WESCON Conv. Record, Part 4, August, 96-104.

CHAPTER 11

MODELLING AND PREDICTION OF QUASIPERIODIC SERIES

Quasiperiodic series can be modelled using SVD and nonlinear modelling or nonlinear transformation and linear modelling. Alternatively periodic decomposition and modelling of periodic components may be used.

11.1 INTRODUCTION

Modelling of nearly periodic time series is quite straight forward and can be done for example using the singular value decomposition based method discussed in Sec.7.6, or using Box and Jenkins method discussed in Sec.4.3. So if a quasiperiodic series can be configured into multiple nearly periodic series through decomposition or transformation, the modelling problem can be simplified; this is the basic concept used for modelling quasiperiodic series in this chapter.

As discussed in Sec.8.2.1, quasiperiodic series can be characterized by

(a) certain periodicity which varies irregularly,

(b) the amplitude over the periodic segments which vary irregularly, and

(c) the absence of any definite repeating pattern.

Three different methods of modelling are presented in this chapter. All the methods model a quasiperiodic series in terms of its constituent periodic components (which are not necessarily sinusoidal) but the way the components are extracted and individually modelled is different. All the methods produce periodic (or pseudo-periodic) models which may be used to produce one period ahead prediction. The first two methods are conceptually similar; the modelling involves singular value decomposition (SVD) coupled with two different ways of incorporating nonlinearity in the model, namely nonlinear transformation and nonlinear modelling using a neural network. The third approach, formulated in structural modelling framework, involves direct decomposition into periodic components.

In the first method, the SV-decomposed data matrix is assumed to constitute a relatively regular part and an irregular part. The regular part is modelled as a nearly periodic series as discussed in Sec.7.8.1. The irregular part is nonlinearly transformed such that a further relatively dominant regular part emerges along with an irregular part, and this method of transformation and segregation is continued until no further regular part can be extracted. Each regular part is linearly modelled and the linear prediction is successively reverse (nonlinearly) transformed to produce additive components of the periodic predictor. This modelling scheme is discussed in Sec.11.2.

The second approach uses the same concept of the separation into regular part and the irregular parts etc. but instead of a linear model a neural network is used to model the selected regular part. This scheme is treated in Sec.11.3.

The third approach is conceptually different from the other two. Here the quasiperiodic series is decomposed into multiple nearly periodic components using SVD. Each component has its own fixed period length and a repeating pattern, which may be differently scaled between the periods. Each nearly periodic component is separately modelled. Sec.11.5 details the modelling scheme based on periodic decomposition. This section is also supported by Appendix 11, where the Singular Value Ratio (SVR) spectrum is introduced, which is used to determine the period length of the most dominant periodic component in a quasiperiodic series.

11.2 MODELLING USING SVD AND NONLINEAR TRANSFORMATION

The basic idea is to model a quasiperiodic process as a combination of component processes, periodic in nature, but belonging to hierarchical levels of nonlinearly transformed spaces, where the period length of each component is the same. This method is applicable to series which show a certain degree of repetitiveness, although both the period length as well as the periodic pattern may vary.

11.2.1 Data Preparation

There are two main ways of arranging the data into a matrix as follows.

(a) The *maximum period length* for the quasiperiodic periodic series is considered to be the row length (n) of the data matrix **X**. The consecutive pseudo-periods are aligned (say with respect to the peak or the trough etc.) into the consecutive rows of **X**, and linearly interpolated data are used for the rows shorter than the maximum length.

(b) The *period length of the most dominant periodic component* present in the data is considered to be the row length (n) of the data matrix **X**. The most dominant periodicity in the series is detected using the SVR spectrum (discussed in Appendix 11). The successive nearly periodic segments in the series are compressed or expanded to the length n as follows.

Let y(1), y(2), ..., y(n*) be the data in a particular segment of the original series, which are to be replaced by the contracted or expanded data set x(1), x(2),...,x(n) where $n \neq n^*$; the transformation is given by

$$x(j) = y(j^*) + (y(j^*+1) - y(j^*))(r_j - j^*), \qquad (11.2.1)$$

where

$$r_j = (j - 1)\,\frac{(n^*-1)}{(n-1)} + 1$$

and j^* is the integer part of r_j. Thus the quasiperiodic series {y(.)} with varying period length is converted into an augmented series {x(.)} having the period length n. The series {x(.)} is aligned into the rows of the data matrix **X**.

The modelling and prediction procedures are the same irrespective of the way the data are arranged. The predicted data have to be reverse transformed with respect to the linear interpolation used or the compression or expansion applied as stated above for producing prediction in the original time domain.

11.2.2 Modelling and Prediction

Let the quasiperiodic sequence be converted into an Mxn data matrix **X** as discussed earlier, and let an mxn data matrix **A**(k) be moving over **X**, where m<M. Define the kth data window **A**(k) by

$$A(k) = \begin{bmatrix} a_{k-m+1}^T \\ a_{k-m+2}^T \\ \vdots \\ a_k^T \end{bmatrix}. \qquad (11.2.2)$$

SVD is performed on A(k) for each value k. If SVD of A(k) produces r dominant singular values,

$$\begin{aligned} A(k) &= USV^T \\ &= ZV^T = \sum_{i=1}^{r} z_i(k)v_i^T(k) + \sum_{i=r+1}^{p} z_i(k)v_i^T(k), \qquad p = \min(m,n) \qquad (11.2.3) \\ &= A_L(k) + A_N(k) \qquad (11.2.4) \\ &= \begin{bmatrix} a_L(k-m+1) \\ a_L(k-m+2) \\ \vdots \\ a_L(k) \end{bmatrix} + \begin{bmatrix} a_N(k-m+1) \\ a_N(k-m+2) \\ \vdots \\ a_N(k) \end{bmatrix} \quad \text{(say).} \end{aligned}$$

Thus $a_L(k)$ and $a_N(k)$ are the last rows of $A_L(k)$ and $A_N(k)$ respectively. $A_L(k)$, which is composed of the dominant decomposition components (in (11.2.3)), is the regular part and the residual $A_N(k)$ is the irregular part of the quasiperiodic process described by A(k).

The objective is to produce the prediction

$$\hat{a}(k+1|k) = \hat{a}_L(k+1|k) + \hat{a}_N(k+1|k). \qquad (11.2.5)$$

$\hat{a}_L(k+1|k)$ is computed from the sequence $\{A_L(k)\}$. To compute $\hat{a}_N(k+1|k)$, successive nonlinear transformation and singular value decomposition is performed. Further details of the prediction procedure follows.

Prediction of the regular component $\hat{a}_L(k+1|k)$

The prediction policy followed for the regular component is the same as discussed in Sec.7.8.1, which is briefly restated below.

(1) When r singular values are dominant, r separate pattern components constitute the regular part of the process.

(2) The i-th dominant component of $A_L(k)$ is given by $(z_i v_i^T)(k)$, $1 \le i \le r$. It has two main parts. The elements of z_i define the scaling factors and the row

$\mathbf{v}_i^T$ defines the pattern.

(3) It is intended to model the series generated by the last (i.e. m-th) element of $\mathbf{z}_i(k)$ for different values of k.

(4) The prediction of $\hat{\mathbf{a}}_L(k+1|k)$, the (m+1)th row of $\mathbf{A}_L(k)$, is computed as

$$(\hat{\mathbf{a}}_L(k+1|k))^T = \sum_{i=1}^{r} \hat{z}_{mi}(k+1|k)\mathbf{v}_i^T(k) \qquad (11.2.6)$$

where $\hat{z}_{mi}(k+1|k)$ is the one-step ahead prediction of $\{z_{mi}(.)\}$ produced at time k.

Remarks

(a) The i-th dominant component of $\mathbf{A}(k)$ and $\mathbf{A}_L(k)$ are the same in (2) above.

(b) The modelling method assumes $\mathbf{v}_i^T$ in (11.2.6) remains unchanged between the windows $\mathbf{A}(k)$ and $\mathbf{A}(k+1)$.

(c) r number of predictors are run simultaneously in (11.2.6), where r may be typically 1 or 2.

Prediction of the irregular component $\hat{\mathbf{a}}_N(k+1|k)$

The fundamental step in the extraction of information from $\mathbf{A}_N(k)$ is to find a suitable nonlinear transformation which enhances the dominance of a periodic component; this results in an increased ratio of the singular values, s_1/s_2, for the transformed matrix compared with that for $\mathbf{A}_N(k)$. The prediction policy can be summarized as follows.

(1) Generate mxn matrix $\mathbf{B}$ through a suitable nonlinear transformation $f_{N1}(.)$ of $\mathbf{A}_N$:

$$\mathbf{B}(k) = f_{N1}(\mathbf{A}_N(k)). \qquad (11.2.7)$$

(2) Perform SVD of $\mathbf{B}$ and segregate the regular and the irregular parts:

$$\begin{aligned}\mathbf{B}(k) &= \mathbf{U}_b\mathbf{S}_b\mathbf{V}_b^T,\\ &= \sum_{i=1}^{r_b} \mathbf{u}_{bi}s_{bi}\mathbf{v}_{bi}^T + \sum_{i=r_b+1}^{p} \mathbf{u}_{bi}s_{bi}\mathbf{v}_{bi}^T, \qquad p=\min(m,n) \qquad (11.2.8)\\ &= \mathbf{B}_L(k) + \mathbf{B}_N(k).\end{aligned}$$

(3) Compute the prediction $\hat{\mathbf{b}}_L(k+1|k)$ due to $\mathbf{B}_L$, the same way as $\hat{\mathbf{a}}_L(k+1|k)$ in (11.2.6):

$$(\hat{\mathbf{b}}_L(k+1|k))^T = \sum_{i=1}^{r_b} \hat{g}_{bi}(k+1|k)\mathbf{v}_{bi}^T(k). \qquad (11.2.9)$$

where r_b are the number of singular values of $\mathbf{B}_L(k)$ that are dominant, and $\hat{g}_{bi}(k+1|k)$ is obtained from $\{\mathbf{B}_L(.)\}$, the same way as $\hat{z}_{mi}(k+1|k)$ is obtained from $\{\mathbf{A}_L(.)\}$ in (11.2.6). $v_{bi}^T(k)$ in (11.2.9) is assumed to remain unchanged between the consecutive windows $\mathbf{B}(k)$ and $\mathbf{B}(k+1)$.

(4) Obtain $\hat{a}_N(k+1|k)$, due to $\mathbf{B}_L(k)$ by reverse nonlinear transformation of $\hat{b}_L(k+1|k)$:

$$\hat{a}_N(k+1|k) = f_{N1}^{-1}(\hat{b}_L(k+1|k)). \qquad (11.2.10)$$

(5) Compute the complete prediction using (11.2.6) and (11.2.10):

$$\hat{a}(k+1|k) = \hat{a}_L(k+1|k) + \hat{a}_N(k+1|k). \qquad (11.2.11)$$

Remarks:

(a) The nonlinear transformation f_{N1} is a one-to-one transformation (i.e. which can be applied individually to each element), e.g., the logarithmic transformation. It is difficult to find the optimal nonlinear transform mathematically, (see for example Tukey, 1957), so a suitable transformation may be determined experimentally. Some transformations will require some preparatory operations, e.g. in case of logarithmic transformation, the positivity of the data has to be ensured.

(b) To decide the suitability of a nonlinear transformation, the effect of the reverse nonlinear transformation of the prediction needs to be taken into consideration. In terms of the singular values, the nonlinear transformation $f_{N1}(.)$ in (11.2.7) should satisfy the condition:

$$(s_{r+1}/s_{r+2}) < (f_{N1}^{-1}(s_{b1}/s_{b2}),$$

where s_{r+1} and s_{r+2} are the singular values of $\mathbf{A}(k)$ and s_{b1} and s_{b2} are the singular values of $\mathbf{B}(k)$.

(c) In (11.2.10), only the information from $\mathbf{B}_L$ in the form of $\hat{b}_L(k+1|k)$ is contained in $\hat{a}_N(k+1|k)$. Further information may be extracted from $\mathbf{B}_N(k)$ of $\mathbf{B}(k)$ as follows: Generate $\mathbf{C}(k)$ through some suitable nonlinear transformation $f_{N2}(.)$ of $\mathbf{B}_N(k)$:

$$\mathbf{C}(k) = f_{N2}(\mathbf{B}_N(k)) = \mathbf{C}_L(k) + \mathbf{C}_N(k), \qquad (11.2.12)$$

and repeat steps similar to (2) to (3) to compute $\hat{c}_L(k+1|k)$, which is reverse transformed to

$$\hat{b}_N(k+1|k) = f_{N2}^{-1}(\hat{c}_L(k+1|k)).$$

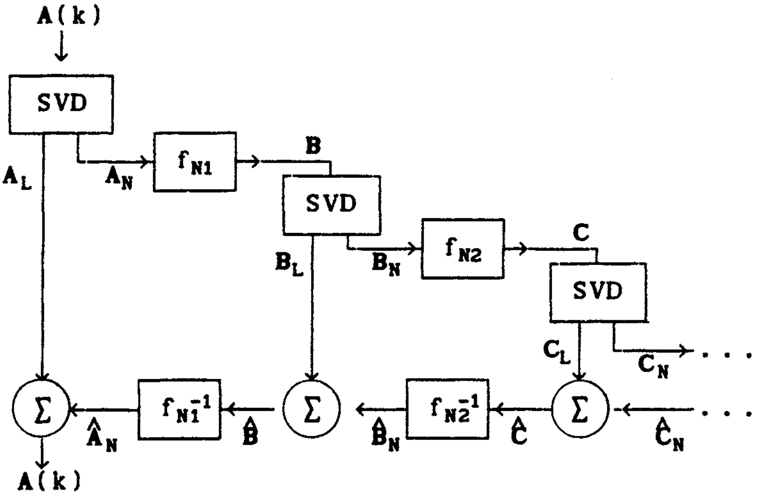


Figure 11.2.1 Model schematic for the quasiperiodic series using SVD and nonlinear transformation.

The sum of $\hat{b}_L(k+1|k)$ and $\hat{b}_N(k+1|k)$ is now reverse transformed in (11.2.10) to compute $\hat{a}_N(k+1|k)$.

Summary of modelling and prediction scheme

A schematic description of the model of the quasiperiodic process is shown in Fig.11.2.1. The model can be summarized as follows.

$$
\begin{aligned}
A(k) &= A_L(k) + A_N(k) \\
&= A_L(k) + f_{N1}^{-1}[f_{N1}(A_N(k))] \\
&= A_L(k) + f_{N1}^{-1}[B(k)] \\
&= A_L(k) + f_{N1}^{-1}[B_L(k) + f_{N2}^{-1}[f_{N2}(B_N(k))]] \\
&= A_L(k) + f_{N1}^{-1}[B_L(k) + f_{N2}^{-1}[C(k)]] \\
&= A_L(k) + f_{N1}^{-1}[B_L(k) + f_{N2}^{-1}[C_L(k) + C_N(k)]],
\end{aligned}
\tag{11.2.13}
$$

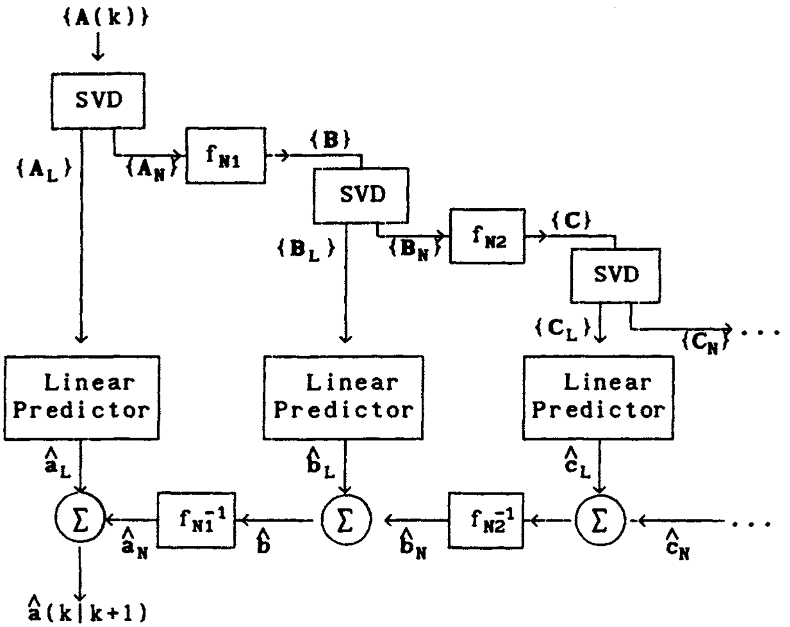


Figure 11.2.2 The one (pseudo-)period ahead prediction scheme

where f_{N1}, f_{N2} etc. are nonlinear transformations; $A_L(k)$, $B_L(k)$, $C_L(k)$ etc are the relatively regular parts of $A(k)$, $B(k)$ and $C(k)$ respectively obtained through SVD. Whether further extraction of information is worth considering will depend on the corresponding signal magnitude of the error component due to $C_N(k)$ in the original time (or spatial) domain, given by

$$\epsilon(k) = A(k) - \hat{A}(k), \tag{11.2.14}$$

where

$$\hat{A}(k) = A_L(k) + f_{N1}^{-1}[B_L(k) + f_{N2}^{-1}[f_{N2}(C_L(k))]]. \tag{11.2.15}$$

The corresponding prediction law is given by

$$\hat{a}(k|k+1) = \hat{a}_L(k|k+1) + f_{N1}^{-1}(\hat{b}_L(k|k+1) + f_{N2}^{-1}(\hat{c}_L(k|k+1))), \tag{11.2.16}$$

where $\hat{a}(k|k+1)$ is the prediction of the (m+1)th row of mxn matrix $A(k)$ at time k. The schematic diagram for the prediction procedure is shown in Fig.11.2.2.

11.2.3 Application Study using the Sunspot Series

Arranging the data

The yearly averaged sunspot data over the years 1700 - 1972 (Appendix 8A) are used in this study. The nearly repetitive (pseudo-periodic) segments of the series are arranged into the consecutive rows of **X**; the segments are considered to start from the lowest points which are aligned into the first column. The period lengths vary between 9 to 14; for the present study the row-length of **X** is chosen as 14. The rows which are shorter in length are appended with linearly interpolated values. **X** works out to be a 23x14 matrix.

Remark
The interpolation of the data is necessary in order to fill up all the positions in the matrix **X** such that SVD analysis can be performed. It has been found that the interpolation scheme does not affect the singular value decomposition appreciably in comparison with that obtained without appending data and using the minimum pattern length (i.e. using n as 9 instead of 14, in the present case). □□

A 4x14 data window **A**(k) is considered moving over the complete data set **X**. The first 20 rows of **X** are used for modelling, and the rest are used for validation through prediction performance.

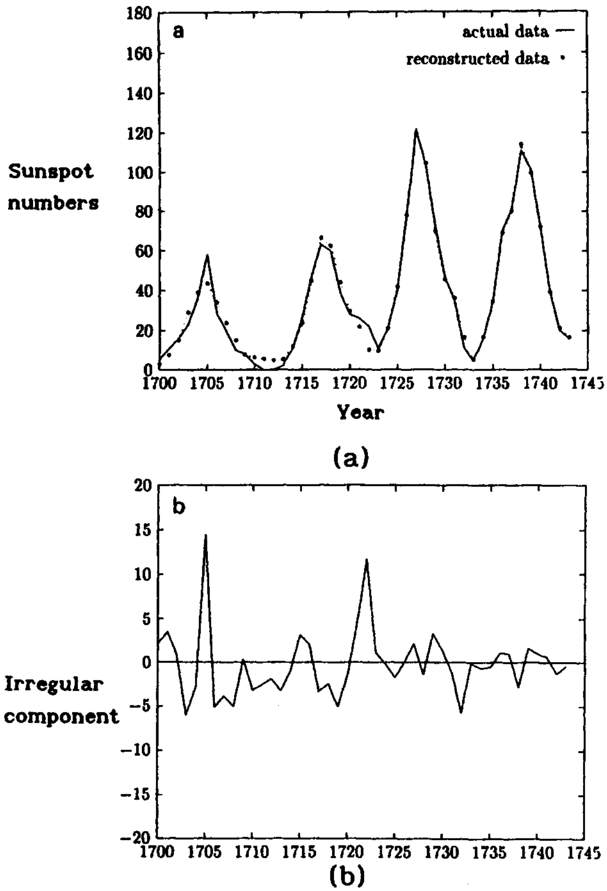
For consecutive positions of **A**(k), SVD is performed on **A**(k). It is observed that $(s_1/s_2)(k)$ is of the order of 7 and that $\mathbf{v}_1(k)$ and $\mathbf{v}_2(k)$ remain fairly unchanged between consecutive windows. So in the model (11.2.3), r = 2. The progressive distribution of the singular values of **A**(k) is shown in Fig.11.2.4.

Reconstruction

The reconstruction of **A**(1) in terms of $\mathbf{A}_L(1)$ is shown in Fig.11.2.3 for the years 1700 to 1743 along with the original sunspot activity series. The interpolated data points are eliminated and have not been shown in the figure. Fig.11.2.4 shows the residual irregular part $\mathbf{A}_N(1)$. As expected, the regular part shows a repetitive pattern, whereas the irregular appears to be random.

Prediction of $\hat{\mathbf{a}}_L(k+1|k)$ from the regular component $\mathbf{A}_L(.)$

The two basic problems are prediction of $\hat{z}_{m1}(k+1|k)$ and



(a)

(b)

Figure 11.2.3 (a) The reconstruction of the sunspot series given by $A_L(1)$, using two most significant singular values of $A(1)$.
———— actual data, ... reconstructed data.
(b) The residual irregular part ($A_N(1)$) after separation of two most significant components $A_L(1)$ from $A(1)$.

$\hat{z}_{m2}(k+1|k)$. Since 20 cycles (pertaining to the years 1700 to 1922) are used for modelling, and since $A(k)$ is of size 4x14, k varies from 1 to 17. So 17 data points are available for modelling each of the sequences $\{z_{m1}(.)\}$ and $\{z_{m2}(.)\}$ for the prediction of the 21st period. Note that z_{m1} and z_{m2} sequences are described by (11.2.3), for example in the present case $z_{m1}(k) = (u_{41}s_1)(k)$, where u_{41} is the 4th element of $u_1(1)$.

Computation of $\hat{z}_{m1}(k+1|k)$

It is found that $\{z_{m1}(.)\}$ can be modelled as an ARIMA (0,1,4) process:

$$\Delta z_{m1}(k) = D(q^{-1})\varepsilon(k) \qquad (11.2.17a)$$

where ε represents white noise, $D(q^{-1})$ is a discrete-time polynomial of order 4:

$$D(q^{-1}) = 1 + d_1q^{-1} + d_2q^{-2} + d_3q^{-3} + d_4q^{-4},$$

and Δ is the unit difference operator.

At each step k, the parameters of (11.2.17a) are estimated and the prediction $\hat{z}_{m1}(k+1|k)$ is computed as

$$\hat{z}_{m1}(k+1|k) = z_{m1}(k) + D(q^{-1})\varepsilon(k), \qquad (11.2.17b)$$

where $\varepsilon(k)$ is estimated as

$$\varepsilon(k) = \Delta z_{m1}(k) - \Delta\hat{z}_{m1}(k|k-1).$$

Computation of $\hat{z}_{m2}(k+1|k)$

It is found that the $\{z_{m2}(.)\}$ process can be modelled by a third-order AR model:

$$F(q^{-1})z_{m2}(k) = \eta(k), \qquad (11.2.18a)$$

where

$$F(q^{-1}) = 1 + f_1q^{-1} + f_2q^{-2} + f_3q^{-3},$$

and $\{\eta(k)\}$ is the noise sequence. At each step k, the parameters of (11.2.18a) are computed, and one-step ahead prediction is produced as

$$\hat{z}_{m2}(k+1|k) = (1-F(q^{-1}))z_{m2}(k). \qquad (11.2.18b)$$

The prediction $\hat{a}_L(k+1|k)$ is obtained from (11.2.6) as

$$(\hat{a}_L(k+1|k) = \hat{z}_{m1}(k+1|k)v_1(k)+ \hat{z}_{m2}(k+1|k)v_2(k). \qquad (11.2.19)$$

Prediction of $\hat{a}_N(k+1|k)$ from the irregular component $A_N(.)$

It is found that logarithmic transformation is a suitable transformation in the present case. A constant value of 60 is added to all the elements of $\{A_N(k)\}$ to ensure that all the elements are positive before transformation. The $\{A_N(k)\}$ sequence is now logarithmically transformed to $\{B(k)\}$ and the SVD of $\{B(k)\}$ is performed for k = 1 to 17. The

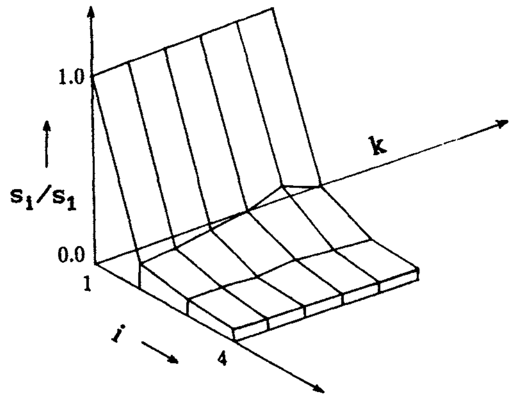


Figure 11.2.4 The progressive distribution of (normalized) singular values of {A(k)} for the sunspot series.

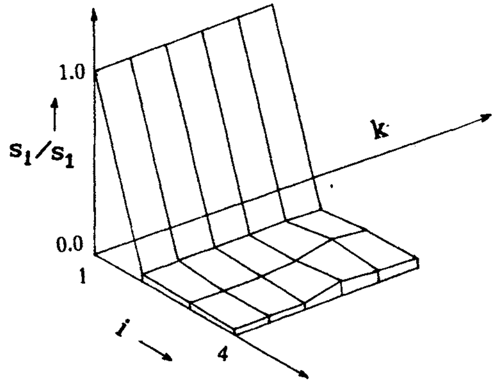


Figure 11.2.5 The progressive distribution of singular values of the log-transformed irregular part {B(k)}.

progressive distribution of the singular values is shown in Fig.11.2.5. It is observed that there is sharp segregation in the distribution with one dominant singular value, whereas before transformation the data appeared to be random as shown in Fig.11.2.3(b).

The prediction $\hat{a}_N(k+1|k)$ is computed using (11.2.8 - 11.2.10). It was observed that only the 1st right singular vector of **B**(.), i.e. v_{b1}^T, remains almost unchanged between consecutive windows of **B**(.); so in (11.2.8), r_b = 1. The

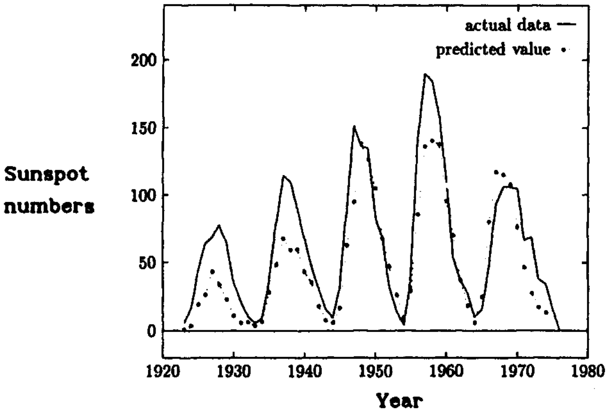


Figure 11.2.6 Prediction (from 1 up to 14 step ahead prediction) of the sunspot series over 21st to 25th (pseudo-)periods from the year 1700; ———— data.

sequence $\{z_{b1}(k)\}$ is modelled as

$$H(q^{-1})z_{b1}(k) = \eta(k), \qquad (11.2.20a)$$

where

$$H(q^{-1}) = 1 + h_1(q^{-1}) + h_2(q^{-2}) + h_3(q^{-3}).$$

The one-step ahead prediction is given by

$$\hat{z}_{b1}(k+1|k) = (1 - H(q^{-1}))z_{b1}(k), \qquad (11.2.20b)$$

where the parameters estimated from (11.2.20a) are used.

The prediction $\hat{b}_L(k+1|k)$ is computed using (11.2.9) as

$$\hat{b}_L(k+1|k) = \hat{z}_{b1}(k+1|k)v_{b1}(k);$$

following (11.2.10) $\hat{b}_L(k+1|k)$ is antilog transformed and a constant value 60 is subtracted from all the elements to obtain the vector $\hat{a}_N(k+1|k)$.

Complete prediction

The complete prediction is obtained using (11.2.16). The results of one period ahead prediction for 5 cycles is shown in Fig.11.2.6.

Remarks: The selection of suitable nonlinear transformation may involve a trial and error approach. The appropriate transformation should enhance the dominance of the prime singular value(s) (see concluding Remarks of Sec.8.4.2).

11.3 MODELLING USING SVD AND NEURAL NETWORK

The method discussed here is structurally similar to that presented in the last section; the difference is that in the individual spaces, instead of a linear model as in Sec.11.2.1, $\{z_{mi}(.)\}$ sequences are modelled by neural networks. Thus nonlinearity is incorporated in the modelling in a new form.

Application

Consider the problem of modelling of the sunspot series discussed in Sec.11.2.3.

In the present case, three layer homogeneous neural networks are used to model the $\{z_{mi}(.)\}$ sequences; each of the hidden nodes and the output node of the network has a threshold level and has sigmoidal nonlinearity (between 0 and 1). The backpropagation algorithm is used to train the network. Detailed discussions on the design of feedforward neural networks is given in Chapter 10.

With reference to Sec.11.2.3, $\{z_{41}(k)\}$ is modelled using a network with $z_{41}(k-1)$, $z_{41}(k-2)$, $z_{42}(k-1)$, $z_{42}(k-2)$ as the inputs, and $z_{41}(k)$ as the output. Initially 7 hidden nodes are used. After an initial stage of training, it is

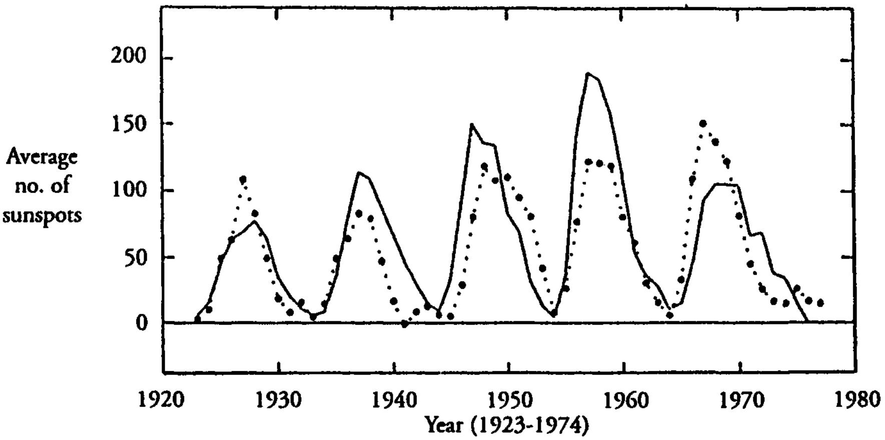


Figure 11.3.1 Prediction of the sunspot series using SVD and neural network models.
——————— actual data, ... predicted values.

found that only 3 hidden layer nodes are necessary. So a 4-3-1 network was retrained to convergence through 10,000 iterations. For modelling of the $\{z_{42}(k)\}$ sequence a network of similar structure was used.

The trained networks are used to produce the predictions $\hat{z}_{41}(k+1|k)$ and $\hat{z}_{42}(k+1|k)$ and $\hat{a}_L(k+1|k)$ is produced from (11.2.19). For $\hat{a}_N(k+1|k)$, the same result as obtained in Sec.11.2.3 is used. the final predictions are produced in Fig.11.3.1.

Remark: For modelling of $\{z_b(k)\}$ sequence in (11.2.20) a nonlinear model can also be used. However, in the present case, as the contribution of this component in the overall prediction is not very significant, only the linear model is considered.

11.4 MODELLING OF A QUASIPERIODIC SERIES THROUGH PERIODIC DECOMPOSITION

11.4.1 Introduction to Periodic Decomposition

A quasiperiodic time series or signal may be decomposed into a number of components each of which is periodic or nearly periodic in nature but *not necessarily sinusoidal*. The quasiperiodic series may be modelled by modelling the individual periodic components separately. The attributes of a constituent periodic component are

(i) a fixed period length, and

(ii) a fixed periodic pattern which may be differently scaled over the periods.

Remark

In the case of Fourier decomposition, each component is *sinusoidal* having a specific frequency, whereas in the case of periodic decomposition, each periodic component may consist of any number of sinusoidal components.

Why periodic decomposition

Often, a quasiperiodic time series or process may be generated from a number of auxiliary processes each producing a periodic or nearly periodic pattern. Frequency domain analysis can decompose the composite signal into constituent sinusoids, whereas the different constituent periodic components may contain overlapping bands of sinusoidal components.

For example, consider the ECG signal, picked up from the abdominal lead on an expectant mother (Sec.14.5.4). It shows the maternal ECG component along with the fetal ECG component. The two components are individually (nearly) periodic but mutually asynchronous. In such cases it is more meaningful to separate the two ECG components instead of analysing the sinusoidal components that make up the composite maternal ECG signal.

The mechanism of periodic *decomposition* and the modelling and prediction of quasiperiodic series through decomposition of periodic components is detailed in this section; the *extraction* of periodic components from composite signals is treated in Sec.14.5.

The mechanism of periodic decomposition

The periodic decomposition involves a two step procedure which is performed successively for extraction of successive periodic components in order of the energy content as follows.

(1) Determine the period length n_1 of the strongest periodic component present in the data series.

(2) Configure the data into a matrix **A** having the row length n_1, perform SVD of **A**: $\mathbf{A} = \mathbf{USV}^T$, and extract the principal periodic component $\mathbf{A}_{p1} = \mathbf{u}_1 s_1 \mathbf{v}_1^T$.

The residual data series is formed from $\{\mathbf{A} - \mathbf{A}_{p1}\}$ and is further decomposed by going through steps (1) and (2). This procedure is continued until no further extraction is possible. If altogether N different periodic components can be extracted, the resulting decomposition is given by

$$\{\mathbf{A}\} = \{\mathbf{A}_{p1}\} + \{\mathbf{A}_{p2}\} + \ldots + \{\mathbf{A}_{pN}\}, \tag{11.4.1}$$

where the components have N specific period lengths.

Remark: If more than one periodic components are detected to have the same period length, they will be mutually orthogonal.

11.4.2 Period Length Estimation for Periodic Components

The period length can be determined, if unknown, using the singular value ratio (SVR) spectrum.

The basic idea for the period length estimation is that if the strongest periodic component in a series is of period length n_1, then a matrix **A**, with row-length n_1, formed using the data will produce a large value of s_1/s_2 on singular value decomposition, where s_1 and s_2 are the first two singular values of **A**.

The basic scheme for the estimation of n_1 is that the data matrix **A** is formed for various choices of row-length n, and the ratio s_1/s_2 is computed for each **A**. The variation of s_1/s_2 against the row-length n, will show a peak at n_1 *and its multiples*; thus n_1 is estimated. The generic term *Singular Value Ratio (SVR) spectrum* is used to refer to the variation of s_1/s_2 against the row-length n; in place of s_1/s_2, s_1^2/s_2^2 or $s_1^2/(s_1^2+...+s_p^2)$, may also be used, where p = rank(**A**). In the present study the ratio s_1^2/s_2^2, referred to as ρ, has been used. For a detailed study of the SVR spectrum, please refer to Appendix 11.

Remark

Since the peaks in the frequency spectrum of the composite series will show the overall contributions of the individual components in different frequencies, the Fourier spectrum cannot serve the same purpose as the SVR spectrum.

11.4.3 Estimation of the Strongest Periodic Component

The estimated periodic pattern primarily belongs to two categories: the principal SV-decomposed periodic pattern and the average energy pattern.

Principal SV-decomposed periodic pattern

Given any $m \times n_1$ matrix **A**, its principal periodic pattern is given by

$$\mathbf{A}_{p1} = \mathbf{u}_1 s_1 \mathbf{v}_1^T = \mathbf{z}_1 \mathbf{v}_1^T, \qquad (11.4.2)$$

where $\mathbf{A} = \mathbf{USV}^T = \mathbf{ZV}^T$, and $\mathbf{u}_1$, $\mathbf{v}_1$ and $\mathbf{z}_1$ are the first columns of **U**, **V** and **Z** respectively. The time series given by $\{\mathbf{A}_{p1}\}$ will have the same repeating pattern given by $\mathbf{v}_1^T$, which will be weighted by the factors $\{z_{j1}\}$ where z_{j1} is the j-th element of $\mathbf{z}_1$ weighing the j-th row of $\mathbf{A}_{p1}$.

The principal pattern series $\{\mathbf{A}_{p1}\}$ can be modelled in terms of $\mathbf{v}_1^T$ and a time series given by the sequence of elements $\{z_{j1}\}$, where the latter can be modelled by a linear model (e.g., an AR model) or a nonlinear model (e.g., a neural network model).

Average energy pattern

The series with average energy pattern is a strictly periodic series where each period contains average row energy of A_{pi}. The average energy pattern is studied in Sec.14.5.2.

The energy contained in the i-th row of A_{pi} is given by

$$\begin{aligned} E_i &= [u_{i1}s_1v_1^T]^T[u_{i1}s_1v_1^T] \\ &= u_{i1}^2[s_1v_1^T]^T[s_1v_1^T]; \end{aligned}$$

hence, using (14.5.1), the average energy periodic pattern is given by

$$\hat{y} = \frac{s_1v_1^T}{\sqrt{m}},$$

where m is the column length of u_1.

Averaged pattern over a moving window

Given an Mxn_i matrix A, an mxn_i data window A′ may be assumed to move over A, where m<M. Conceptually this arrangement is the same as in Appendix 7A.2. There will be M-m+1 such windows. For each position of A′, the prime component as in (11.4.2) is computed. An average of the candidate patterns obtained from the different windows is computed for each period and thus the complete periodic sequence of period length n_i is constructed. This scheme may produce better result for dynamic series.

11.4.4 Implementation Considerations

(1) *Prime steps in implementation*

The periodic decomposition has two prime elements: the determination of period lengths and the estimation of the periodic components. The period lengths can be uniquely determined through procedures discussed in Sec.11.4.2 but the estimation of the periodic components largely depends on the type of data and the applications concerned.

(2) *Zero-meanness of components*

In (11.4.2), if v_1^T is zero-mean, the corresponding time series $\{A_{pi}\}$ will also be zero-mean.

(3) *Nature of periodic patterns*
The change in sign of the elements of $\mathbf{u}_1$ or $\mathbf{z}_1$ will result in the inversion of the pattern in the corresponding time series. This phenomenon results from the fact that from an algebraic point of view, there is no time series information associated with $\mathbf{A}_{p1}$; the time series information is imparted through the modelling of the series $\{z_{j1}\}$. There are two possibilities:
(a) The series, $\{z_{j1}\}$ can be modelled even if the elements of $\{z_{j1}\}$ have different signs.
(b) If the reversal of the pattern is not acceptable, one may proceed as follows. (i) A nonzero constant value may be added to the data to make it non-negative; the consequent decomposition should be free from inversions of periods. The decomposition will be meaningful if the energy in the constant mean is much less than the same in the rest of the data. (ii) Alternatively, a periodic series with an average energy pattern may be used; this approach is preferrable for non-prime periodic components.

(4) *Chances of modelling noise*
With successive extraction of periodic components, the residual becomes more and more noisy; in such cases care should be taken not to model the noise as periodic component(s).

(5) *SVR spectrum analysis*
(a) The SVR spectrum will also show peaks at multiples of the wave lengths of interest (as discussed in Appendix 11); so the corresponding shortest period length is the one desired. (b) SVR spectrum analysis requires only the singular values and hence will not be computation-intensive, if the singular vectors are not computed. The typical figures for the number of operations (flops) for the computation of S and of U,S,V are $(2mn^2 + 2n^3)$ and $(4m^2n + 22n^3)$ respectively for an mxn matrix (Golub and Van Loan, 1989, p.239).

11.4.5 Application Examples

The periodic decomposition can be applied both for separation of constituent periodic component(s) in a composite series as well as for the modelling and prediction of any quasiperiodic series through the periodic component(s).

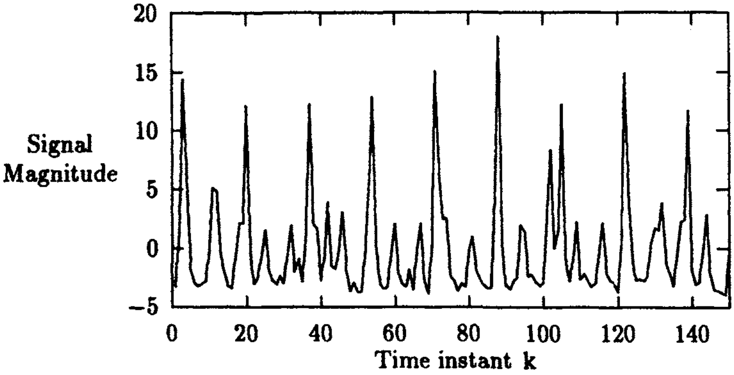


Figure 11.4.1 The composite series ⟨y(k)⟩ to be decomposed.

Example 11.4.5(1) Decomposition of a composite signal into periodic components

Three perfectly periodic components y_{17}, y_7 and y_{30} of periodlengths 17, 7 and 30 respectively are added along with white Gaussian noise of signal-to-noise ratio 5 with respect to the weakest periodic component y_{30}, forming the composite signal ⟨y(k)⟩ shown in Fig.11.4.1. The decomposition is performed as follows.

(1) The SVR spectrum is computed for row lengths varying between 5 to 50. SVR spectrum (Fig.11.4.2(a)) shows peaks at row lengths of 17 and 34, indicating presence of a component with period length 17. ⟨y(k)⟩ is formed into a matrix **A** of row length 17, which is SV-decomposed, and the component $\langle\hat{y}_{17}(k)\rangle$ is extracted using (11.4.2).

(2) The SVR spectrum of the residual sequence $\langle y(k)-\hat{y}_{17}(k)\rangle$ is computed for row lengths varying between 5 and 50. The SVR spectrum (Fig.11.4.2(b)) shows peaks at row lengths of 7, 14, 21 etc. So the data $\langle y(k)-\hat{y}_{17}(k)\rangle$ are formed into a matrix of row length 7, which is SV-decomposed, and the component $\langle\hat{y}_7(k)\rangle$ is extracted.

(3) The SVR spectrum of the subsequent residual sequence $\langle y(k)-\hat{y}_{17}(k)-\hat{y}_7(k)\rangle$ is now formed; its SVR spectrum (Fig. 11.4.2(c)) for row lengths varying between 5 to 70 shows peaks at row length 30 and 60; so the data are arranged into a matrix of row length 30 and using SVD the periodic component of period length 30, $\langle\hat{y}_{30}(k)\rangle$, is extracted.

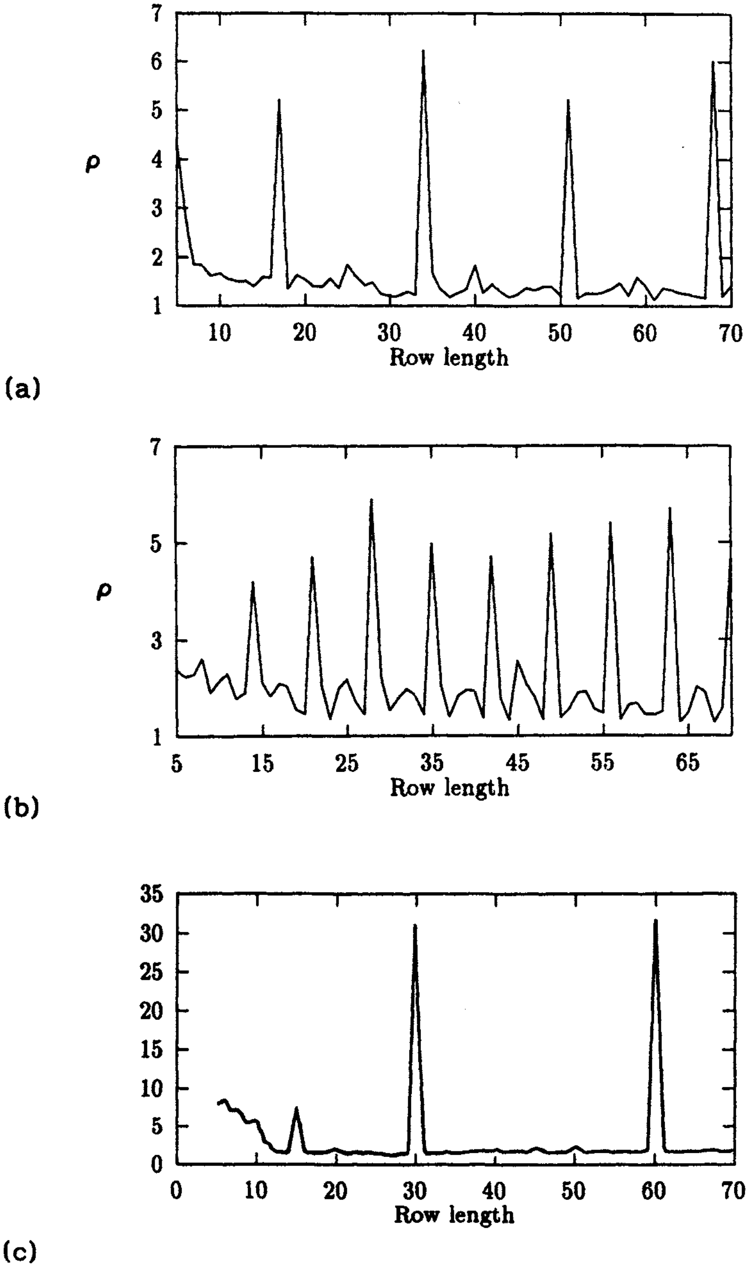


Figure 11.4.2 (a) SVR spectrum of the composite signal $\{y(k)\}$ shown in Fig.11.4.1,

(b) SVR spectrum of the residual sequence $\{y(k)-\hat{y}_{17}(k)\}$,

(c) SVR spectrum of the subsequent residual sequence $\{y(k)-\hat{y}_{17}(k)-\hat{y}_7(k)\}$.

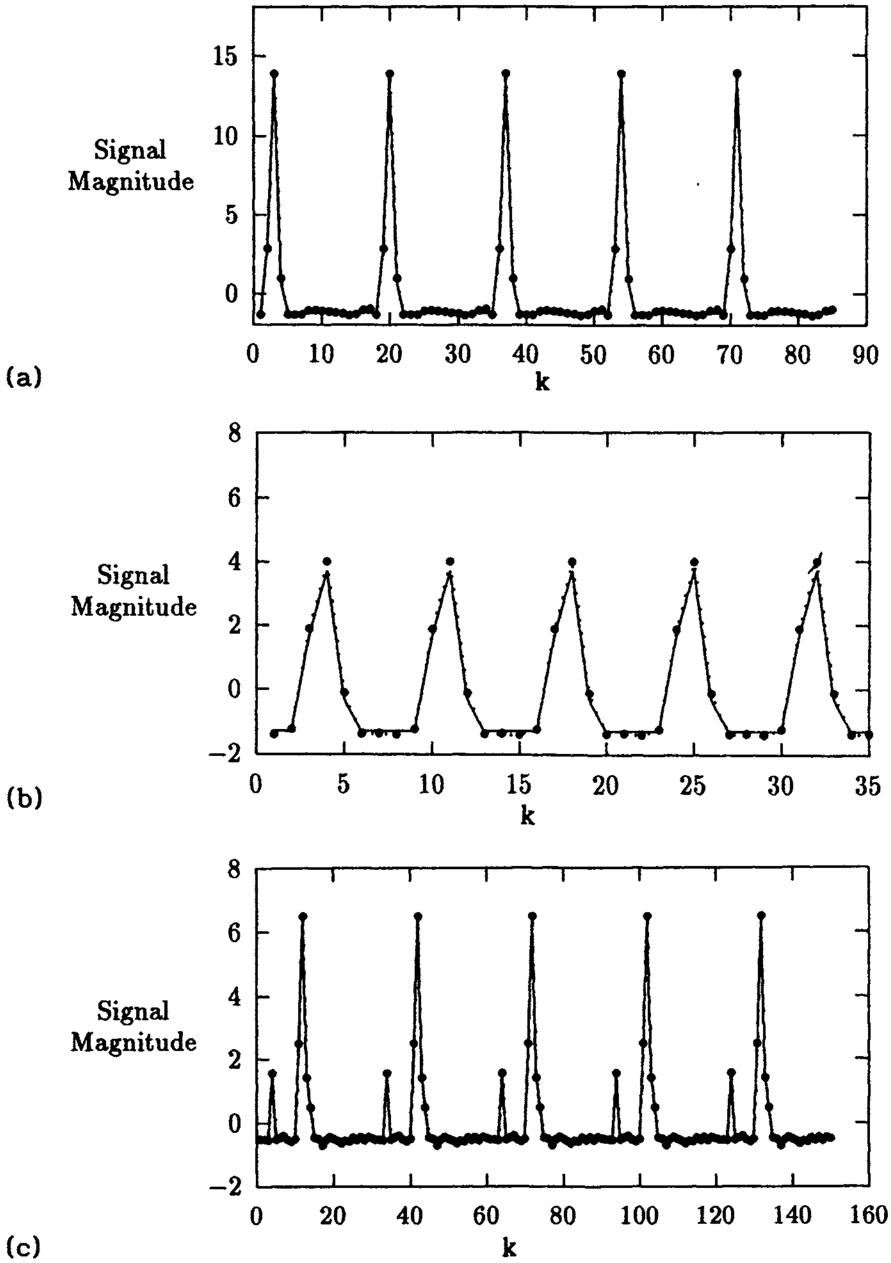


Figure 11.4.3 The decomposed periodic components of $\{y(k)\}$. ———— actual data, ... estimated data
(a) The actual component $\{y_{17}(k)\}$ and its estimate,
(b) the actual component $\{y_7(k)\}$ and its estimate,
(c) the actual component $\{y_{30}(k)\}$ and its estimate.

The resulting residual sequence $\{y(k)-\hat{y}_{17}(k)-\hat{y}_7(k)-\hat{y}_{30}(k)\}$ is found to be of too small magnitude to be of interest. The extracted (estimated) periodic components are shown in Fig.14.4.3.

Remark
It can be shown (Kanjilal and Palit, 1995) that the periodic decomposition is possible even in the case of components having nonintegral period lengths.

Case Study: Modelling and long term prediction of the sunspot series

Decomposition

The yearly averaged sunspot data for the years 1700–1920 are used for modelling. To construct the SVR spectrum, a data window size 4 is used; the median of the available values for the ratio ρ $(=s_1^2/s_2^2)$ is plotted against varying row length. SVR spectrum of the series (Fig.11.4.4) shows the first periodic component to be of period length 11, which is extracted using (11.4.2). The residual series is further analysed by SVR spectrum and a peak is detected at the period length of 10; the corresponding periodic component is also extracted. SVR spectrum on the subsequent residual series does not show any conspicuous peak. The estimated periodic components of the sunspot series are shown in Fig.11.4.5.

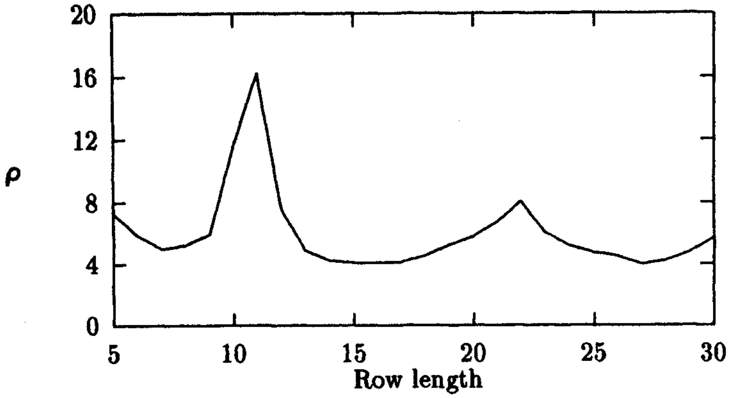


Figure 11.4.4 SVR spectrum of the sunspot series

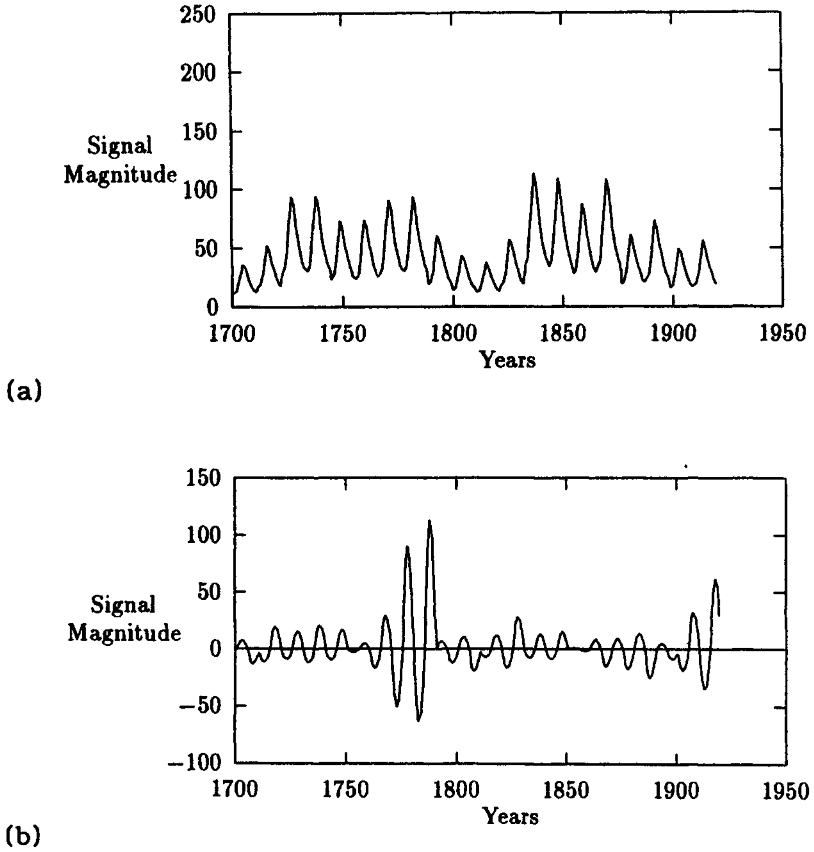


Figure 11.4.5 (a) The strongest periodic component in the sunspot series with period length 11,
(b) the second periodic component with period length 10.

Prediction

For each periodic component given by $z_1 v_1^T$ in (11.4.2), one period ahead prediction is given by $\hat{z}_{(m+1)1} v_1^T$, where $\hat{z}_{(m+1)1}$ is the predicted (m+1)th element of the m-vector z_1; the elements of z_1 (say, {x(k)}) may be modelled as an AR process.

In this particular case, the elements of z_1 corresponding to the period length 11 are modelled as

$$x(k) = -128.659 - 0.542x(k-1) + 0.344x(k-2),$$

and the same corresponding to the period length 10 are modelled as

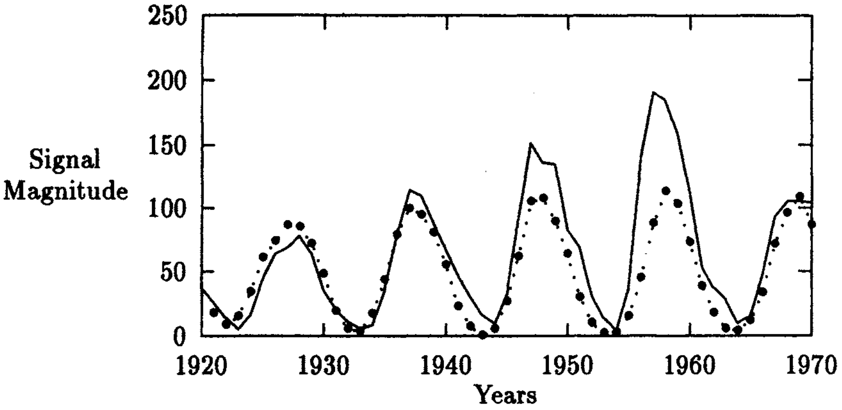


Figure 11.4.6 The prediction of the sunspot series for the period 1921 to 1975.

$$x(k) = -24.738 - 0.759x(k-1) + 0.597x(k-2) - 0.129x(k-3).$$

In both the cases, the terminating point for the periodic components are considered to be the same (which is 1920 here). One period ahead prediction of both the series are computed and are replicated to the year 1970. Fig.11.4.6 shows the prediction of the series from 1921 to 1970 using data up to 1920. The MSE for the actual prediction period 1921-1931 is 129.28 per sample.

11.5 CONCLUSIONS

Modelling of quasiperiodic series in terms of periodic components has been presented, a periodic component being characterized by a repetitive pattern associated with a scaling factor. Three different approaches have been considered. The first two approaches are applicable to data series which exhibit a certain degree of repetitiveness of the pattern. In both the cases the period length is assumed to remain unchanged and interpolated data are used within the sequence wherever necessary for (near-)alignment of consecutive (pseudo-)periods for modelling purposes. The third approach involves periodic decomposition of any series with or without any conspicuous periodicity.

In the first approach nonlinear transformation and linear modelling is used. The quasiperiodic series is first singular value decomposed into a relatively regular part and a residual irregular part. The irregular part is nonlinearly transformed such that after transformation again a relatively regular part emerges as dominant and so on. The relatively regular parts individually represent nearly periodic series. Since SV-decomposition of a periodic series produces a pattern component and scaling factors for the pattern component to represent the consecutive periods, each periodic series is modelled as having a constant pattern and the scaling factors are modelled as a linear series. Usually the original time domain periodic component is the strongest component, and the amount of information which can be sensibly modelled from the relatively irregular part is very much dependent on the correct choice of the nonlinear transformation applied. This is one of the disadvantages of this method.

In the second approach, the scaling factor sequence is modelled nonlinearly using a neural network. Here, the main emphasis is on better modelling of the prime relatively regular part only. Since nonlinearity is permitted in the model of the scaling factor sequence, it is expected that the representativeness of modelling will improve in this case, which is confirmed by the illustrative examples.

The modelling of a quasiperiodic process through periodic decomposition is by far most straightforward. The periodic components can have any period length which is determined and need not be assumed. The scaling factors for the consecutive periods in case of each component can be modelled linearly or nonlinearly. It is expected that since the underlying characteristic of the process is modelled in terms of constituent periodic components, the quality of long term (that is multiperiod in this case) prediction will also be good.

The subject treated in this chapter is an area of recent development, and it is expected that further work will lead to more improved models and prediction results.

REFERENCES

Remarks: The characteristics of quasiperiodic and chaotic processes are treated in [8], and nonlinear transformation is studied in [1,2,11]; these subjects also feature in Chapter 8 of this book. Modelling and prediction of quasiperiodic processes are detailed in [9,10]. The subjects presented in Secs. 11.2 to 11.4 also appear in the papers [4, 5, 6, 7].

[1] Atkinson, A.C. (1985): *Plots, Transformations and Regression*, Oxford Univ. Press, Oxford.

[2] Box, G.E.P., and D.R. Cox. (1964): 'An analysis of transformations', *Journal of Royal Stat. Soc.*, B, 26, 211-252.

[3] Golub, G.H., and C.F. Van Loan (1989): *Matrix Computations*, 2nd edn., The Johns Hopkins Univ. Press, Baltimore.

[4] Kanjilal, P.P., and S. Palit (1994): 'The singular value decomposition - applied in the modelling and prediction of quasiperiodic processes', *Signal Processing*, 35(3), 257-267.

[5] Kanjilal, P.P., S. Palit, and G.P. Rao (1994): 'On the modelling and prediction of quasiperiodic signals using SVD and neural networks', IFAC Symp. on System Identification, SYSID '94, July, Copenhagen.

[6] Kanjilal, P.P., and S. Palit (1995): 'On multiple pattern extraction using singular value decomposition', *IEEE Trans. on Signal Processing*, 43(6), 1536-1540.

[7] Kanjilal, P.P., and S. Palit (1995): 'On the modelling and prediction of time series using singular value decomposition and neural networks', *Computers in Electrical Engineering*, to appear.

[8] Parker, T.S., and L.O. Chua (1987): 'Chaos: a tutorial for engineers', *Proc. IEEE*, 75(8), 982-1008.

[9] Subba Rao, T., and M.M. Gabr (1984): *Introduction to Bispectral Analysis and Bilinear Time Series Models*, Lecture Notes in Statistics, No.24, Springer-Verlag, Berlin.

[10] Tong, H. (1983): *Threshold models in non-linear time series analysis*, Lecture Notes in Statistics, No.21, Springer-Verlag, New York.

[11] Tukey, J.W. (1957): 'On the comparative anatomy of transformations', *Annals of Math. Stat.*, 28, 602-632.

CHAPTER 12

PREDICTIVE CONTROL (Part-I): INPUT-OUTPUT MODEL BASED

Predictive control aims at obtaining the predicted performance of the process as specified.

12.1 INTRODUCTION

Given any input-output process, a mathematical model of the process can be formulated from the available measurements and informations, and a suitable control input to be applied to the process can be computed such that a specified performance criterion is satisfied. Appropriate control input is expected to result in the process output reaching the desired set point. Process control can be difficult when the process itself, as represented by the model, is difficult to control, or when the information available to the controller is imprecise or incorrect. Some features which make a real-life process difficult to control are as follows:

(a) the order of the process may not be known,

(b) the process may be dynamic but the operational data available on the input and output may not be rich and may not reveal the salient process characteristics,

(c) the time delay between the input and the output may not be fixed or known,

(d) the process may be open-loop unstable,

(e) the process may be nonminimum-phase in nature.

The first three features make the correct identification and parameter estimation of the process difficult; the last three features demand particular attention in the choice of the cost criterion which the control law must optimize. In addition there may be unknown disturbances acting on the process. A successful control strategy has to perform sensibly irrespective of the inaccuracies, inconsistencies and difficulties stated above.

The design of the controller is influenced by the process model available but the quality of control largely depends on the performance criterion optimized.

The performance criteria may concern the desired closed-loop controller characteristics, as in case of the pole-placement controller, or it may concern a cost criterion (or cost-function) to be minimized. The cost criterion minimized can be based either

(a) on the past performance, or
(b) on the predictive performance.

The traditional PI (i.e. proportional plus integral) control belongs to the former category while the predictive control belongs to the latter; the predictive control performs better particularly for processes with time delay because the future dynamics of the process can be more correctly defined in this case. Again the cost on predictive performance can be based either

(a) on a single step cost function, or
(b) on a multistep cost function,

where the single or multistep refers to the number of time steps over which the cost function is optimized. The conventional self-tuning control (STC) belongs to the former category; the latter is the class of controllers, known as the *Long-range predictive controllers*. 'Long-range' refers to the prediction horizon being extended over multiple steps into the future; the control law takes into account the implication of the control action over this predictive horizon. The attractive feature of long range predictive control (LRPC) methods is that they possess better stability and robustness properties than the non-LRPC methods, and can perform satisfactorily irrespective of wrong prior assumptions about the process and its environment. Different designs are possible for predictive controllers; these are characterized by the process model, the prediction scheme, and the cost function optimized.

This chapter describes some of the popularly used predictive control methods for linear systems. A real-life process is usually dynamic in nature, and it works in a stochastic environment; so it is necessary for the controller to be adaptive. Sec.12.2 introduces STC, which is one of the widely used adaptive control methods; STC automatically tunes its parameters in a stochastic environment to obtain desired performance of the closed loop system. The cost minimized is a single-step function of the deviation of the predicted output from the desired value at a specific point in future, which is usually the process time-delay ahead (the time-delay being the time between the application of an input and its response in the output). In practice, STC can provide stable control provided certain prior assumptions about the process remain valid. However, improper choices of time-delay or model order (which are quite likely in complex dynamical processes) can easily destabilize STC. LRPC schemes can be designed to provide satisfactory control under the same circumstances.

The most important feature of an LRPC is the minimization of a multistep cost-function, which is a quadratic function of the deviation of the predicted output from the desired output over multiple steps (or a horizon) in future. In this chapter three popular classes of LRPC algorithms are studied, namely the pulse-response model based LRPC, the step-response model based LRPC and the Generalized Predictive Control algorithm which is CARMA or CARIMA model based LRPC. The different LRPC methods have certain generic structural features which are introduced in Sec.12.3. The main difference between the different designs of LRPCs is due to the process models used. The algorithmic details of the three different LRPC designs are presented in Sec.12.4, Sec.12.5 and Sec.12.6. The performance of the LRPC schemes largely depend on the choice of the different design parameters in the cost-function optimized; this subject is probed in Sec.12.7. Again, success of a predictive control method is largely dependent on numerically and computationally robust implementation, which is discussed in Sec.12.8.

This chapter is supported by two Appendices. Appendix 12A presents a brief introduction to the area of systems and controls. Appendix 12B studies the Smith predictor, which is one of the early approaches to predictive control in a deterministic environment.

12.2 SELF-TUNING CONTROL

12.2.1 Basic Concepts

An adaptive control scheme can be broadly considered as a combination of an on-line estimation method for the process parameters, and a controller design procedure. There are two widely used configurations of adaptive controllers:

(1) *Self-organizing adaptive controllers* recursively identify the process, and formulate the control strategy aiming at optimal performance.

(2) *Model reference adaptive controllers* try to achieve a closed-loop system performance similar to that of a reference model by recursive adaptation of the controller parameters.

The self-organizing adaptive controllers may be categorized into different configurations based on the performance evaluation technique and the controller design procedure:

(a) *Dual controllers* are those that perform the dual simultaneous functions of realizing the desired performance and reducing the model uncertainty. These are designed on the basis of available measurements as well as the future observation programme and the associated statistics.

(b) *Non-dual controllers* are based on the present and the past information only. As a result, the rate at which the uncertainty about random process variables is reduced, is independent of the control action. Non-dual controllers may be further classified into certainty equivalent and cautious controllers.

(i) *Certainty equivalent controllers* ignore the fact that the estimated parameters, which are used to design the controller, are not the true ones. On the other hand,

(ii) *Cautious control* implies modification of the control action in recognition of the uncertainty associated with the estimated parameters.

Self-tuning control (STC), which is one of the most widely used adaptive control strategies, belongs to a class of non-dual, certainty equivalent type self-organizing controllers.

STC is applicable to processes with constant or slowly varying parameters. A basic schematic diagram of the application of STC is shown in Fig.12.2.1. At every sampling instant, the process parameters are estimated, using a recursive estimation method, and the estimated parameter values are used to compute the control law, which optimizes a prespecified cost criterion. Thus the uncertainty in the

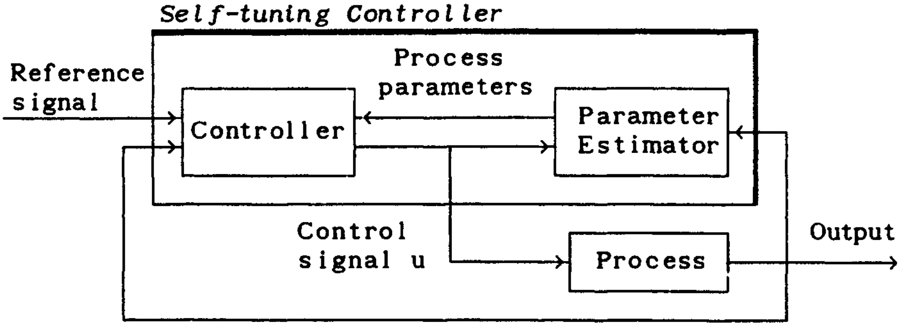


Figure 12.2.1 Self-tuning control scheme

estimated parameters is ignored and the control algorithm uses them as if they are true. Thus the parameter estimation and the design of the control law emerge as two separate activities. The idea of separating the parameter estimation from the controller design is known as the *separation principle*. Although the algorithm starts with limited knowledge of the process, in the course of the recursions with the progression of time, the process parameters tend to converge to the true values, and the controller approaches an optimal state. This conforms to the *self-tuning property*, which implies eventual convergence of the control law to one that could be designed, if the true process model were known.

The concept of self-tuning dates back to Kalman (1958), who estimated the plant parameters from an on-line least squares algorithm and the estimates were used to derive a dead-beat control law at every sampling instant. The self-tuning principle was later revived by Peterka (1970), who incorporated stochastic features in the formulation. The subsequent reported works of Åström and Wittenmark (1973), and Clarke and Gawthrop (1975), boosted research and practical applications of self-tuning controllers.

12.2.2 Control Algorithms

Process model

Consider an input-output model of the process in a stochastic environment

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + \alpha(k), \tag{12.2.1}$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_nq^{-n},$$
$$B(q^{-1}) = b_o + b_1q^{-1} + \dots + b_nq^{-n},$$

d is the maximum time-delay between the control input u and the measured value of process output y; α is the noise or uncertainty in the model. There can be different representations of α as follows.

(a) $\alpha(k) = e(k)$, (12.2.2)

where {e(k)} is an uncorrelated random sequence.

(b) $\alpha(k) = C(q^{-1})e(k)$, (12.2.3)

where $C(q^{-1})$ is assumed to be a stable but unknown polynomial,

$$C(q^{-1}) = 1 + c_1q^{-1}+ \dots + c_nq^{-n}.$$

(c) $\alpha(k) = T(q^{-1})e(k)$, (12.2.4)

where $T(q^{-1})$ is a discrete-time noise observer polynomial:

$$T(q^{-1}) = 1 + t_1q^{-1} + \dots + t_rq^{-1}.$$

(d) $\alpha(k) = T(q^{-1})e(k)/\Delta$, (12.2.5)

where $\Delta = 1-q^{-1}$, the unit difference operator.

Note that (12.2.2) leads to a controlled AR model, (12.2.3) or (12.2.4) leads to a CARMA model, and (12.2.5) leads to a CARIMA model of the process.

Cost criterion

The minimized cost function is given by

$$J = E\{((w(k)-y(k+d))^2 + \lambda(Qu(k))^2)|k\} \tag{12.2.6}$$

where w is the target set-point for the output to reach, λ is a scalar cost factor; Q, which may be a transfer function, is typically chosen as, $Q = \Delta = 1-q^{-1}$. The objective is to determine u(k) which minimizes the cost J.

Remarks

(a) When w(k) is constant, the *controller* is called the *regulator.*

(b) When the costing on the control (λ or Q) in J in (12.2.6) is too large, the control will resemble open-loop control, and when the costing on the control is zero, the control becomes the minimum variance control.

(c) The cost function (12.2.6) is expressed as ensemble average because the process model is stochastic in nature. When the parameters of the process model remain unchanged, the cost criterion J yields the same optimal strategy as the one minimizing J':

$$J' = \{\frac{1}{N}\sum_{i=1}^{N}\{(w(k) - y(k+i))^2 + \lambda(Qu)^2\}\}.$$

Control Law

Assume the noise structure (12.2.5). Introduce the identity

$$T(q^{-1}) = E(q^{-1})A(q^{-1})\Delta + (q^{-d})F(q^{-1}), \qquad (12.2.7)$$

where the degree of E, $\delta E = d-1$ and $\delta F<\delta(\Delta A)$,

$$E(q^{-1}) = 1 + e_1e^{-1} + \dots + e_{d-1}q^{-d+1}, \qquad (12.2.8)$$

$$F(q^{-1}) = f_o + f_1e^{-1} + \dots + f_{n-2}q^{-n+2}. \qquad (12.2.9)$$

From (12.2.1) and (12.2.5), dropping the index (q^{-1}) for simplicity,

$$EA\Delta y(k+d) = EB\Delta u(k) + ETe(k+d).$$

Using (12.2.8-12.2.9),

$$Ty(k+d) = Fy(k) + G\Delta u(k) + ETe(k+d), \qquad (12.2.10)$$

where

$$G(q^{-1}) = E(q^{-1})B(q^{-1})$$
$$= g_o + g_1q^{-1} + \dots + g_{n+d-1}q^{-n-d+1}.$$

Since $E(q^{-1})$ is of order (d-1), Ee(k+d) will have terms e(k+d),..., e(k+1) etc. which will be independent of y and u terms in Fy(k) and $G\Delta u(k)$. Hence the d-step ahead predictor is given by

$$\hat{y}(k+d|k) = Fy_f(k) + G\Delta u_f(k), \qquad (12.2.11)$$

where y_f and u_f are the filtered signals:

$$y_f(k) = y(k)/T(q^{-1}),\ u_f(k) = u(k)/T(q^{-1}),$$

and F and G are estimated polynomials.

The prediction error is given by

$$\tilde{y}(k+d|k) = Ee(k+d).$$

Since

$$\frac{d\hat{y}(k+d|k)}{du(k)} = g_o,$$

the cost in (12.2.6) is minimized for

$$(\hat{y}(k+d|k) - w(k))g_o + \lambda Qu(k) = 0. \tag{12.2.12}$$

The control law is obtained as

$$u(k) = \frac{w(k) - Fy_f(k) - \left(\frac{G}{T} - g_o\right)\Delta u(k) + g_o u(k-1)}{g_o + \frac{\lambda Q}{g_o}}. \tag{12.2.13}$$

Characteristic Features

(a) Closed-loop properties

Equation (12.2.13) may also be expressed as

$$u(k) = \frac{w(k) - \frac{F}{T}y(k)}{\frac{G\Delta}{T} + Q'}, \qquad Q' = \frac{\lambda Q}{g_o}. \tag{12.2.14}$$

The closed-loop equation obtained by substituting for u(k) from (12.2.14) in the process model is given by

$$y(k) = \frac{B}{B+Q'A}\,w(k-d) + \frac{\left(G + \frac{Q'T}{\Delta}\right)e(k)}{B + Q'A}. \tag{12.2.15}$$

Thus even if the process is a nonminimum-phase process (i.e. $B(q^{-1})$ having roots outside the unit circle), with suitable costing Q′ on the control, the closed-loop roots will be stable. Q′ can shift the closed-loop roots from those of $B(q^{-1})$ (i.e. for Q′ being low) towards those of $A(q^{-1})$ (i.e. for Q′ being high). Thus if the open-loop process is stable (i.e. $A(q^{-1})$ being stable), the closed-loop process will also be stable. Again, improper choice of the time-delay d may also produce b_0 as low or zero, where appropriate control costing can ensure stable control.

The present control algorithm is not applicable for processes which are both nonminimum-phase as well as open-loop unstable.

(b) Model reference features

Model reference features may be introduced into the self-tuning control algorithm by defining an auxiliary output ψ as

$$\psi(k) = P(q^{-1})y(k), \qquad P(q^{-1}) = \frac{P_n(q^{-1})}{P_d(q^{-1})},$$

and devising a predictive control law u(k) such that $\psi(k+d)$ reaches w(k), which implies y(k+d) reaching $w(k)/P(q^{-1})$, the filtered set point; in other words the output follows the model $M(q^{-1}) = 1/P(q^{-1})$. The choice of P having a steady-state value of unity: $P(q^{-1})|_{q=1} = 1$, ensures offset-free control. Typically $P(q^{-1}) = (1-0.6q^{-1})/0.4$.

Remarks

(a) *Control weighting*: The control weighting in the cost function (12.2.6) can serve two basic purposes: (i) to produce stable control for nonminimum-phase plants and (ii) to generate moderate or acceptable (also called *detuned*) control action. For offset-free control, $Q(q^{-1})$ should be a transfer function with zero steady-state gain, the simplest choice being $Q(q^{-1}) = 1-q^{-1}$.

(b) *Noise observer polynomial* $T(q^{-1})$: In most practical applications a first order $T(q^{-1})$ suffices. $1/T(q^{-1})$ effectively acts as a low-pass filter for the data. Since the time differencing Δ, acts as a high-pass filter, $\Delta/T(q^{-1})$ effectively acts as a band-pass filter on the data (e.g., in (12.2.17)).

Implementation aspects

The two broad approaches for the design of the self-tuning controller are: (i) the explicit or indirect method and (ii) the implicit or direct method. These methods differ in the way the controller parameters are estimated. The implementation requires $T(q^{-1})$, $Q(q^{-1})$, λ and w(k) to be known. A summary of the explicit and the implicit methods follows.

Explicit approach

(1) The parameters of the process model (12.2.1) are estimated using a recursive parameter estimator.

(2) The parameters of $E(q^{-1})$ and $F(q^{-1})$ are computed using (12.2.7).

(3) The control u(k) is determined using (12.2.13).

Implicit approach

The controller parameters can be directly estimated from (12.2.10) but the input data in (12.2.10) are incremental whereas the output data are positional; this imbalance may affect parameter estimation. To alleviate this problem introduce the identity:

$$F(q^{-1}) = T(q^{-1}) + \Delta F'(q^{-1}). \tag{12.2.16}$$

So (12.2.10) can now be expressed as

$$y(k+d)-y(k) = \frac{\Delta F'(q^{-1})y(k)}{T} + \frac{G\Delta u(k)}{T} + Ee(k+d). \tag{12.2.17}$$

The implicit method can proceed as follows:

(1) The controller parameters are directly estimated from (12.2.10) using a recursive parameter estimation method.

(2) The control u(k) is determined using (12.2.13).

Practical aspects

(a) Robust control requires robust estimation. If the data are not rich but rather steady, the estimator becomes prone to blow-up of the covariance matrix. In such cases it is necessary to bypass the parameter estimation stage. For further discussions in robust parameter estimation see Sec.3.4.2.

(b) The control law discussed assumes a linear process. In practice, nonlinearity may be present in various forms. For example in the case of an industrial process, the actuator, which is used to implement a change in the control input, may have nonlinear characteristics; such types of nonlinearities should be taken into account separately. It is necessary to ensure that the control action proposed is actually exerted.

Remark: These practical aspects concern *all adaptive controllers.*

Example 12.2.2 Compute self-tuning control law for the process given by

$$(1 - 0.8q^{-1})y(k) = 0.2u(k-2) + (1-0.6q^{-1})e(k),$$

minimizing the cost given by (12.2.6) with Q = 1, and given values of λ and $w(k)$.

The process model used is

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k).$$

Following (12.2.7–12.2.13), the key equations for the present problem are as follows (the polynomial index (q^{-1}) is dropped for simplicity).

(i) $C = EA + q^{-d}F, \qquad T(q^{-1}) = C(q^{-1}),$

with $E = 1 + e_1q^{-1}$, $\delta F < \delta A$ or $F = f_o$.

(ii) $Cy(k+d) = Fy(k) + EBu(k) + ECe(k+d).$

(iii) $\hat{y}(k+d|k) = Fy(k)/T + Gu(k)/T, \qquad G = EB.$

(iv) $$\frac{\partial J}{\partial u(k)} = -(w(k) - \hat{y}(k+d|k))g_o + \lambda u(k) = 0$$

$$= -w(k) - Fy_f(k) + \left(G + \frac{\lambda T}{g_o}\right)u_f(k), \qquad y_f = y/T,\ u_f = u/T.$$

(v) $$u_f(k) = \frac{w(k) - Fy_f(k) - (G-g_o)u_f(k) - \frac{\lambda}{g_o}(T-1)u_f(k)}{g_o + (\lambda/g_o)}.$$

In the present case

$$A(q^{-1}) = 1-0.8q^{-1},\ B(q^{-1}) = 0.2,\ C(q^{-1}) = (1-0.6q^{-1}),$$

and d = 2. E and F are solved using the identity

$$(1-0.6q^{-1}) = (1+e_1q^{-1})(1-0.8q^{-1}) + q^{-2}f_o.$$

So, $e_1 = 0.2$, $f_o = 0.16$, and

$$G(q^{-1}) = (1+0.2q^{-1})0.2 = 0.2+0.04q^{-1}.$$

Hence

$$(1-0.6q^{-1})y(k+2) = 0.16y(k) + (0.2+0.04q^{-1})u(k)$$
$$+ (1+0.2q^{-1})(1-0.6q^{-1})e(k+2).$$

That is

$$y(k+2) = [0.16y_f(k) + (0.2+0.04q^{-1})u_f(k)] + (1+0.2q^{-1})e(k+2),$$

where $y_f(k) = y(k)/(1-0.6q^{-1})$ and $u_f(k) = u(k)/(1-0.6q^{-1})$;

note that the noise term outside the parenthesis is uncorrelated to the data at time k. Hence

$$u_f(k) = \frac{w(k) - 0.16y_f(k) - 0.04u_f(k-1) + \frac{\lambda}{0.2}(0.6u_f(k-1))}{0.2 + (\lambda/0.2)},$$

from which the desired control u(k) can be computed.

12.2.3 Controller as Operator-guide: An Example

Many industrial processes work on manual control or fixed parameter control like the PID control. Plant managers not being fully conversant with the capabilities of adaptive control schemes, are often reluctant to permit trials of adaptive controllers in their plants. One possible compromise is to go for adaptive prediction, and off-line control strategy, where on-line plant data are used, and the control is computed but is not applied directly to the plant; the control is used as an operator-guide. An industrial application of such a scheme is presented here.

Strand speed control in iron-ore sintering

One important part of iron-ore sintermaking is the on-strand process (see Sec.5.6.1), where sintering actually takes place. Proper sintering requires the strand to be driven at a certain speed, such that the on-strand process of sintering is complete. Too high a strand speed causes generation of weak sinter and a large amount of return-fines which have to be recycled; too low a strand speed results in a drop in production and uneconomically high sinter strength. The trials reported here were conducted at the Redcar Sintering Plant of British Steel Corporation in 1982.

At the time of the trial, the conventional strand-speed control system was not working, and a manual control was in use, with the operator making occasional changes in the strand speed with the aim of achieving a reasonably steady waste gas temperature (WG Temp). Here the strand speed is treated as the input and the WG Temp is treated as the output.

The objective of the plant trial was

(a) to produce multistep prediction of WG Temp, and

(b) to present to the operator the computed value of the

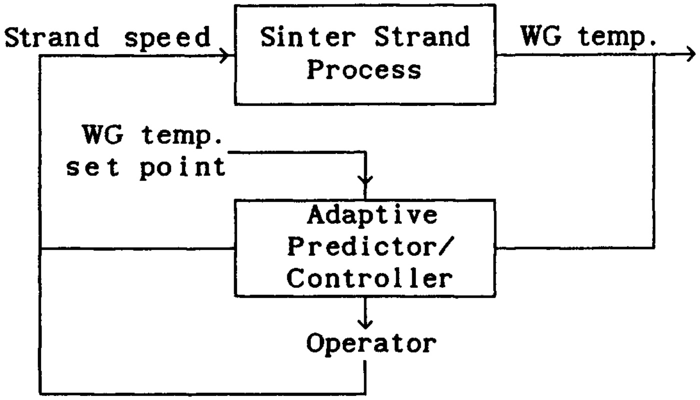


Figure 12.2.2 The sinter strand control scheme with adaptive predictor/controller as operator guide.

predictive control to be treated as the desired strand speed (see Fig.12.2.2).

Since frequent changes in strand speed are undesirable due to the integrated nature of the process (see Fig.5.6.1), the operators were advised by the managenent to change the strand speed according to the predictive control advice, every 1/2 hr, if necessary.

An ARMA model of the process was used; the time delay between the input (strand speed) and the output (WG Temp) was found to be 8 min. from the recorded historical data. The sampling time was chosen as 2 min. A control strategy of the generalized minimum variance category was used to compute the control law.

At each sampling time, the 8 min ahead prediction of the WG Temp was produced as shown in Fig.12.2.3(a). The corresponding (implementable) speed control advice and the actual strand speed profile are shown in Fig.12.2.3(b). It is seen that the WG Temp remained close to the desired range of 145°-150°, except when the control advice had been completely ignored, e.g. around 19 hrs and around 23 hrs.

Remark

Although the controller has been used only as an operator guide, it should be possible to use the controller directly on an on-line basis.

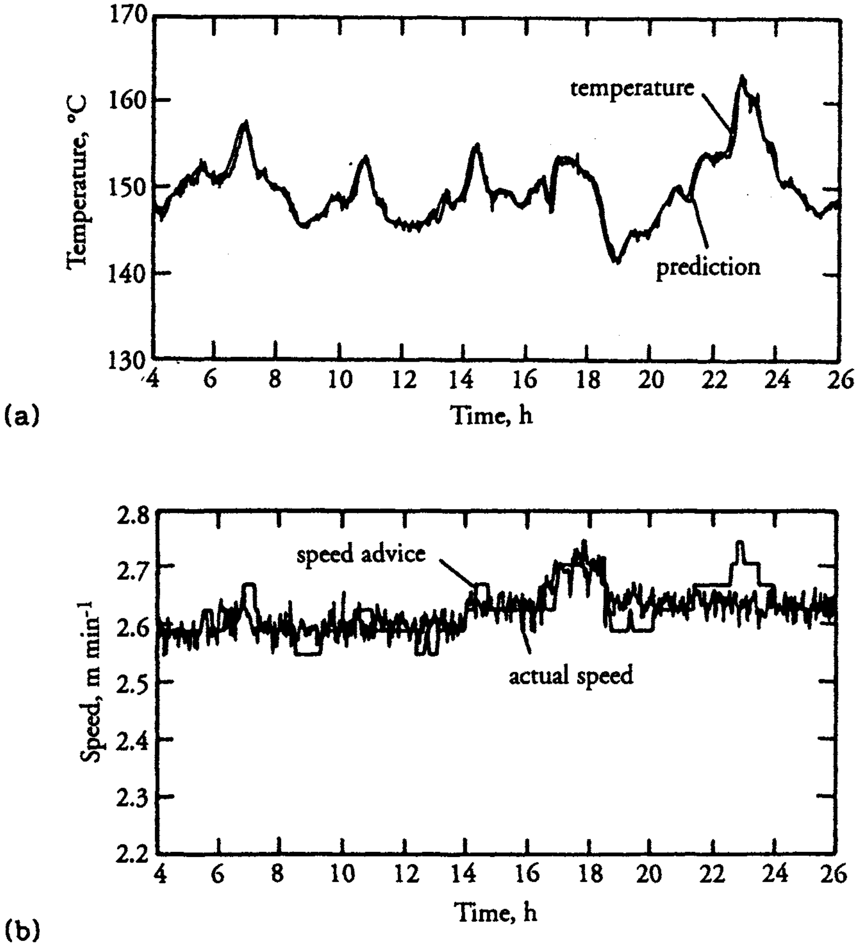


Figure 12.2.3 (a) 8-minute ahead prediction of the WG temperature in the sintering process (plant trial). (b) The strand speed advice and the actual strand speed corresponding to WG temperature shown in (a).

12.3 LONG RANGE PREDICTIVE CONTROL

12.3.1 Introduction

The long range predictive control (LRPC) methods are characterized by the process model used and the multistep cost function optimized. LRPC broadly falls into three categories: (i) pulse response model based, (ii) step response model based, and (iii) the Generalized Predictive Control class which are CARIMA model based. In each of these categories different designs exist, many of which have been successfully used in practice.

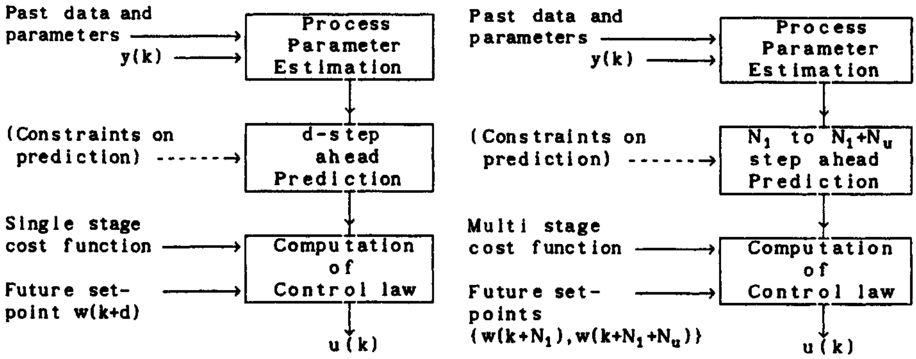
IDCOM (IDentification COMmand) due to Richalet *et al* (1978) is pulse response model based; the *Model Algorithmic Control* (Rouhani and Mehra, 1982) also belongs to this category. The step response model based design is due to Cutler and Ramaker (1980), who called it *Dynamic Matrix Control* (DMC). Clarke and Zhang (1987) modified IDCOM and DMC by incorporating an integrating noise structure in the process model for natural elimination of offsets; this model structure is used in the present study. The Generalized Predictive Control (GPC) due to Clarke *et al* (1987) unifies the different design features for LRPC into one algorithm. The interest in LRPC is mainly because most LRPC designs can produce stable and robust control irrespective of the difficult real-life problems of nonminimum-phasedness, varying time-delay, unknown model order etc.

In self-tuning control discussed in the last section, the control designed at every sampling instant is aimed at driving the output to the desired set-point at a specific time step in future. On the other hand, in LRPC, the control computed at every sampling instant is intended to maintain the output at the desired set-point over certain multiple time steps (or a horizon) in future. Instead of a single-step cost function as in STC, LRPC is based on the minimization of a multistep cost function. The ultimate result is a comparatively superior control performance of LRPC in terms of robustness and stability. The structural difference between STC and LRPC schemes is shown in Fig.12.3.1.

12.3.2 The Generic Structure

LRPCs are characterized by
(a) the process model used, and
(b) the multistep cost function minimized.

The specification of the cost function includes
(a) the horizon in the future over which the cost function is minimized for control calculations,


Past data and parameters
y(k)
Process Parameter Estimation
(Constraints on prediction)
d-step ahead Prediction
Single stage cost function
Computation of Control law
Future set-point w(k+d)
u(k)
Past data and parameters
y(k)
Process Parameter Estimation
(Constraints on prediction)
N₁ to N₁+Nᵤ step ahead Prediction
Multi stage cost function
Computation of Control law
Future set-points {w(k+N₁),w(k+N₁+Nᵤ)}
u(k)

(a) Self-tuning control (b) LRPC

Figure 12.3.1 Comparison between the control law computation schemes of self-tuning control and LRPC.

(b) the type of control increments penalized and the cost on the control increments, and

(c) the future set-point sequence etc.

Fig.12.3.2 shows the typical movements of the output in response to a certain control input sequence, given the desired set-point sequence.

Cost function

The prime objective is to minimize the squared error ε between the predicted output ($\hat{y}$) and the set-point (w) over a specified horizon in future with minimum control effort. The generic cost function is given by

$$J = \underline{\varepsilon}^T\underline{\varepsilon} + \lambda[\Delta u]^T[\Delta u], \qquad (12.3.1)$$

where $\underline{\varepsilon}$ is the predicted error vector, and Δu is the incremental control input:

$$\Delta u = [\Delta u(k)\ \Delta u(k+1)\ \ldots\Delta u(k+N)]^T,$$

$$\underline{\varepsilon} = [\varepsilon(k+1)\ \varepsilon(k+2)\ \ldots\ \varepsilon(k+N)]^T;$$

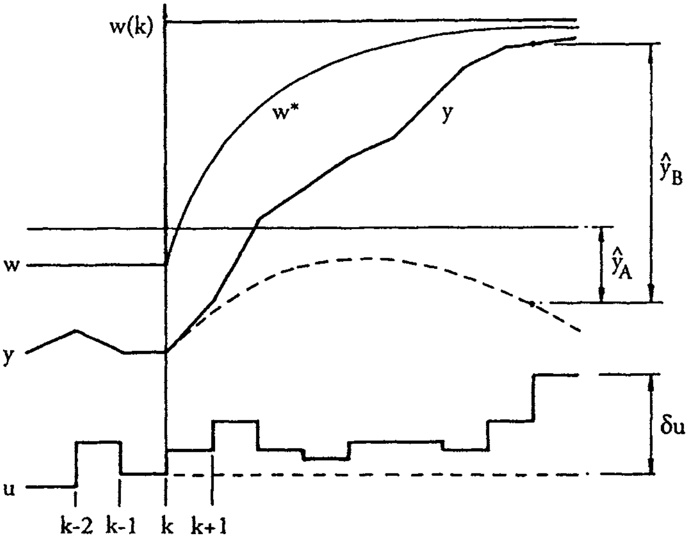


Figure 12.3.2 Typical profiles of the output (y), the set-point (w) and the control input (u); $\hat{y}_A$ is output response due to unchanged control input u(k−1), $\hat{y}_B$ is additional response due to additional control input δ_u, w^* is low-pass filtered set-point trajectory.

$\varepsilon(k+N) = w(k+N) - \hat{y}(k+N|k)$, N being the length of the predictive horizon.

The control law is computed through the following steps: modelling and parameter estimation, prediction of the output, and computation of the control u(k); the latter two steps are summarized below, while the parameter estimation is treated in Sec.12.3.2.

Output prediction

The two basic aspects of output prediction that concern LRPC designs are the horizon over which prediction is performed, and the known and the unknown components of the predicted output.

The output y(k) is predicted over the specified horizon $(k+N_1)$ to $(k+N_2)$. Ideally $N_1 = d$, the time delay, but in the present discussions a default value of $N_1 = 1$ is used.

The prediction $\{\hat{y}(k+i|k),\ i=1 \text{ to } N\}$ has two components, as follows:

$$\begin{aligned}\hat{y} &= [\hat{y}(k+1|k)\ \hat{y}(k+2|k)\ \dots\ \hat{y}(k+N|k)]^T \\ &= [\hat{y}_A(k+1|k)\ \hat{y}_A(k+2|k)\ \dots\ \hat{y}(k+N|k)]^T + \\ &\qquad\qquad [\hat{y}_B(k+1|k)\ y_B(k+2|k)\ \dots\ \hat{y}(k+N|k)]^T \\ &= \quad \hat{y}_A \quad + \quad \hat{y}_B \qquad \text{(say)}, \end{aligned} \tag{12.3.2}$$

where $\hat{y}_A$ is the known, and $\hat{y}_B$ is the unknown component at time k, the elements of $\hat{y}_B$ being functions of unknown control components: $\Delta u(k)$, $\Delta u(k+1)$,... etc. $\hat{y}_B$ is of the generic form

$$\hat{y}_B = G\Delta u = \begin{bmatrix} g_o & & & 0 \\ g_1 & g_o & & \\ \vdots & \vdots & \ddots & \\ g_{N-1} & g_{N-2} & \cdots & g_o \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix}, \tag{12.3.3}$$

where the elements of **G** are functions of the estimated process (or controller) parameters.

Control law

Following (12.3.3),

$$\underline{\varepsilon} = \mathbf{w} - \hat{y} = \mathbf{w} - \hat{y}_A - G\Delta u. \tag{12.3.4}$$

Again, following (12.3.1) and (12.3.4), Δu minimizing the cost function J is given by

$$\Delta u = [G^TG + \lambda I]^{-1}G^T[\mathbf{w} - \hat{y}_A], \tag{12.3.5}$$

where **w** specifies the future set points:

$$\mathbf{w} = [w(k+1)\ w(k+2)\ \dots\ w(k+N)]^T.$$

The objective is to determine u(k), which comes from the first element of Δu.

Set point sequence

Usually the future values of the set point are not known. So the set point is assumed to remain constant at w(k) over the time {k to (k+N)}. If actual values for w(k+1), w(k+2) etc. are known (e.g., in case of robotic movements or in case of flights of space crafts), these values may be used. If the step changes in the set point {w(k)-w(k-1)} are too large, undesirably high values of u may be required to force y to reach the set point, which can be avoided by considering a low-pass filtered set point trajectory w* (see Fig.12.3.2),

$$w^*(k+i) = \frac{(1-\alpha)w(k+i)}{1 - \alpha q^{-1}},$$

where $0<\alpha<1$ and $w^*(k) = w(k)$.

12.4 LRPC: PULSE RESPONSE MODEL BASED

Pulse response model has been discussed in Sec.2.4. Consider the model

$$y(k) = H(q^{-1})u(k) + \frac{e(k)}{\Delta}, \tag{12.4.1}$$

where $\{e(k)\}$ is uncorrelated random sequence, and $\{e(k)/\Delta\}$ models the nonstationary noise; $H(q^{-1})$ is assumed to be given by

$$H(q^{-1}) = h_1q^{-1} + \dots + h_Mq^{-M}.$$

The parameters of $H(q^{-1})$ can be estimated using the recursive least squares method on the incremental model

$$\Delta y(k) = H(q^{-1})\Delta u(k) + e(k)$$

$$= h_1\Delta u(k-1) + h_2\Delta u(k-2) +\dots+ h_M\Delta u(k-M) + e(k).$$

Output prediction

As stated in (12.3.2), consider the predicted output over the horizon (k+1) to (k+N) being composed of two components:

$$\hat{y} = \hat{y}_A + \hat{y}_B.$$

Here (a) $\hat{y}_A$ is output response due to that component of the future control input sequences $\{u(k+i), i\ge 0\}$ which remains unchanged at $u(k-1)$, that is with $\Delta u(k) = \Delta u(k+1) =\dots= 0$, and (b) $\hat{y}_B$ is the response due to the additional components in the control input sequences $\{u(k+i)-u(k-1), i\ge 0\}$. So the component vector $\hat{y}_A$ is known but $\hat{y}_B$ is unknown with $\hat{y}_B(k)=0$. The concept of the two components of the predicted output is schematically shown in Fig.12.3.2.

(i) Prediction $\hat{y}_A$

Following (12.3.9) prediction based on the estimated parameters is given by

$$\begin{aligned}\Delta\hat{y}_A(k+i) &= \hat{y}_A(k+i|k) - \hat{y}_A(k+i-1|k)\\ &= h^T\Delta u_A(k+i-1), \qquad i = 1,2,\ldots,N, \qquad (12.4.2)\end{aligned}$$

where $\mathbf{h} = [h_1\ h_2\ \ldots\ h_N]^T$, and

$$\Delta u_A(k+i-1)$$
$$= [\underbrace{\Delta u(k+i-1), \Delta u(k+i-2), \ldots, \Delta u(k)}_{\text{i terms = 0 (by hypothesis)}}, \Delta u(k-1), \ldots, \Delta u(k+i-N)]^T.$$

(ii) Prediction $\hat{y}_B$

$$\begin{aligned}\Delta\hat{y}_B(k+i) &= h^T\Delta u_2(k+i-1)\\ &= h^T[\Delta u(k+i-1), \Delta u(k+i-2), \ldots, \Delta u(k),\\ &\qquad \underbrace{\Delta u(k-1), \ldots, \Delta u(k+i-N)]^T}_{\text{(N-1) terms = 0}}.\end{aligned}$$

Since $\hat{y}_B(k) = 0$,

$$\begin{aligned}\Delta\hat{y}_B(k+i) &= \Delta\hat{y}_B(k+i) - \hat{y}_B(k)\\ &= \Delta\hat{y}_B(k+i) + \Delta\hat{y}_B(k+i-1) + \ldots + \Delta\hat{y}_B(k+1)\\ &= h^T[(u(k+i-1)-u(k-1)), (u(k+i-2)-u(k-1)),\\ &\qquad \ldots, (u(k)-u(k-1)), 0, \ldots, 0]^T\\ &= h^T[\delta u_i, \delta u_{i-1}, \ldots, \delta u_{i-M+1}]^T, \qquad (12.4.3)\end{aligned}$$

where $\delta u_i = u(k+i-1) - u(k-1)$.

Example:

$$\begin{aligned}\Delta\hat{y}_B(k+2) &= \Delta\hat{y}_B(k+2) + \Delta\hat{y}_B(k+1)\\ &= \hat{y}_B(k+2) - \hat{y}_B(k+1) + \hat{y}_B(k+1) - \hat{y}_B(k)\\ &= h_1(u(k+1) - u(k)) + h_2(u(k) - u(k-1))\\ &\qquad + h_1(u(k) - u(k-1)) + 0\\ &= h_1(u(k+1) - u(k-1)) + h_2(u(k) - u(k-1)).\end{aligned}$$

□□

Hence, the prediction error vector is given by

$$\underline{\varepsilon} = \mathbf{w} - \hat{y}_A - \hat{y}_B,$$

that is

$$\begin{bmatrix} \varepsilon(k+1) \\ \varepsilon(k+2) \\ \vdots \\ \varepsilon(k+p) \end{bmatrix} = \begin{bmatrix} w(k+1) \\ w(k+2) \\ \vdots \\ w(k+p) \end{bmatrix} - \begin{bmatrix} \hat{y}_A(k+1) \\ \hat{y}_A(k+2) \\ \vdots \\ \hat{y}_A(k+p) \end{bmatrix} - \begin{bmatrix} h_1 & & & 0 \\ h_2 & h_1 & & \\ \vdots & & \ddots & \\ h_p & h_{p-1} & \cdots & h_1 \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \\ \vdots \\ \delta u_p \end{bmatrix}$$

$$= \mathbf{w} - \hat{\mathbf{y}}_A - \mathbf{H}[\delta \mathbf{u}_B]. \qquad (12.4.4)$$

Control Algorithm

Consider the cost function

$$J = \underline{\varepsilon}^T \underline{\varepsilon} + \lambda[\delta u_B]^T[\delta u_B], \qquad (12.4.5)$$

where λ is a positive scalar weight. This cost function is minimized for the control law

$$\delta \mathbf{u}_B = [\mathbf{H}^T\mathbf{H} + \lambda \mathbf{I}]^{-1}\mathbf{H}^T[\mathbf{w} - \hat{\mathbf{y}}_1]. \qquad (12.4.6)$$

The desired control is given by

$$u(k) = u(k-1) + \delta\hat{u}_1, \qquad (12.4.7)$$

where $\delta\hat{u}_1$ is the first element of $[\delta u_B]$, computed from (12.4.4).

12.5 LRPC: STEP RESPONSE MODEL BASED

Consider the step response model of the process

$$y(k) = S(q^{-1})u(k) + e(k)/\Delta \qquad (12.5.1)$$

where $\{e(k)\}$ is uncorrelated random sequence, and $\{e(k)/\Delta\}$ models the nonstationary noise; $S(q^{-1})$ is given by

$$S(q^{-1}) = s_1q^{-1} + \ldots + s_Mq^{-M}. \qquad (12.5.2)$$

Output prediction

The problem in using (12.5.1) directly for prediction is the nonconvergence of the $\{s_i\}$ sequence. Instead consider the $\hat{y}_B$ components in (12.4.2) which can be expressed in terms of the s_i parameters of the step response model as follows.

Consider a special case: N = 3, for which

$$(H+\delta u_2)_{N=3} = \begin{bmatrix} h_1 & 0 & 0 \\ h_2 & h_1 & 0 \\ h_3 & h_2 & h_1 \end{bmatrix} \begin{bmatrix} u(k) - u(k-1) \\ u(k+1) - u(k-1) \\ u(k+2) - u(k-1) \end{bmatrix}$$

$$= \begin{bmatrix} h_1 & 0 & 0 \\ h_2 & h_1 & 0 \\ h_3 & h_2 & h_1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) + \Delta u(k) \\ \Delta u(k+2) + \Delta u(k+1) + \Delta u(k) \end{bmatrix}$$

$$= \begin{bmatrix} h_1 & 0 & 0 \\ h_2+h_1 & h_1 & 0 \\ h_3+h_2+h_1 & h_2+h_1 & h_1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \end{bmatrix}$$

$$= \begin{bmatrix} s_1 & 0 & 0 \\ s_2 & s_1 & 0 \\ s_3 & s_2 & s_1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \end{bmatrix},$$

using (12.3.11). Therefore, $\hat{y}_B$ in (12.4.2) can be expressed as

$$H[\delta u_2] = \begin{bmatrix} s_1 & & & \\ s_2 & s_1 & & 0 \\ \vdots & \vdots & \ddots & \\ s_N & s_{N-1} & \cdots & s_1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix}$$

$$= S[\Delta u]. \tag{12.5.3}$$

Hence, following (12.4.3), the prediction error vector $\underline{\varepsilon}$ can be restated as

$$\underline{\varepsilon} = \mathbf{w} - \hat{y}_A - S[\Delta u]. \tag{12.5.4}$$

Control law

Consider the cost function

$$J = \underline{\varepsilon}^T \underline{\varepsilon} + \beta[\Delta u]^T[\Delta u], \tag{12.5.5}$$

where β is the scalar cost on the control increments. This cost function is minimized for the control law

$$\Delta u = [S^T S + \beta I]^{-1} S^T[\mathbf{w} - \hat{y}_1]. \tag{12.5.6}$$

The desired control is given by

$$u(k) = u(k-1) + \Delta\hat{u}(k),$$

where $\Delta\hat{u}(k)$ is the first element of the vector Δu in (12.5.6).

12.6 GENERALIZED PREDICTIVE CONTROL

Consider a CARIMA model of the process:

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + C(q^{-1})\frac{e(k)}{\Delta}, \qquad (12.6.1)$$

where $\{e(k)\}$ is uncorrelated random sequence, and $\{e(k)/\Delta\}$ models the nonstationary noise; the time delay between the input and the output is assumed to be at least 1.

Output prediction

The prediction of the output when the process is represented by a CARIMA (or ARIMAX) model is discussed in Sec.5.2.1; some basic steps for design are restated here.

Consider the explicit prediction approach. The p-step ahead prediction is given by

$$\hat{y}(k+p|k) = F_p(q^{-1})y(k) + G_p(q^{-1})\Delta u(k+p-1),$$

where

$$G_p(q^{-1}) = E_p(q^{-1})B(q^{-1}) = g_o + g_1q^{-1} + g_2q^{-2} + \ldots\text{etc.},$$

and the parameters of E_p and F_p are obtained from the identity

$$C(q^{-1}) = E_p(q^{-1})\Delta A(q^{-1}) + q^{-p}F_p(q^{-1}); \qquad (12.6.2)$$

the degree of E_p, $\delta E_p = p-1$, and $\delta F_p \leq \delta A-1$.

The parameters of $A(q^{-1})$ and $B(q^{-1})$ are estimated from the process model (12.6.1.). For each value of p, the first p-1 parameters of E_p are the same as the p-1 parameters of E_{p-1}; the p-th parameter of E_p and the parameters of F_p are computed from (12.6.2). Again, the first p parameters of G_p: $g_o, g_1, \ldots, g_{p-1}$, will remain unchanged as p is incremented. The recursive estimation of the parameters of E_p and F_p are discussed in Appendix 5A.

The multistep prediction is given by

$$\hat{y}(k+i|k) = f_{1o}y(k) + f_{11}y(k-1) + \ldots$$
$$+ g_o\Delta u(k+i-1) + g_1\Delta u(k+i-2) + \ldots$$

f_{1o} etc. are the parameters of F_p obtained from i-th solution of the identity (12.6.2).

In (12.6.1), if the time delay is d instead of 1, the multistep prediction will be given by

$$\hat{y}(k+i|k) = f_{1o}y(k) + f_{11}y(k-1) + \ldots$$
$$+ g_o\Delta u(k+i-d) + g_1\Delta u(k+i-d-1) + \ldots$$

Example: If i = 5,

$$\hat{y}(k+5|k) = (f_{5o}y(k) + f_{51}y(k-1) + \ldots)$$
$$+ g_o\Delta u(k+4) + g_1\Delta u(k+3) + g_2\Delta u(k+2) + g_3\Delta u(k+1) + g_4\Delta u(k)$$
$$+ (g_5\Delta u(k-1) + g_6\Delta u(k-2)),$$

where the terms within the parantheses are known and the rest are unknown. Thus

$$\hat{y}(k+i|k) = \hat{y}_A(k+i|k) + \hat{y}_B(k+i|k),$$

where $\hat{y}_B$ component of the predicted output $\hat{y}$ is a function of $\Delta u(k)$, $\Delta u(k+1)$, ... etc., which are unknown, whereas $\hat{y}_A$ component is known. So the output vector $\hat{y}$ can be predicted as

$$\hat{y} = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N|k) \end{bmatrix} = \hat{y}_A + \begin{bmatrix} g_o & & & \\ g_1 & g_o & & 0 \\ \vdots & \vdots & \ddots & \\ g_{N-1} & g_{N-2} & & g_o \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix}$$

$$= \hat{y}_A + G[\Delta u] \qquad \text{(say)}, \qquad (12.6.3)$$

where $\hat{y}_A$ and $G[\Delta u]$ represent all the known and the unknown quantities respectively. The prediction error vector $\underline{\varepsilon}$ is given by

$$\underline{\varepsilon} = w - \hat{y} = w - \hat{y}_A - G[\Delta u]. \qquad (12.6.4)$$

Control law

The control law is derived based on the minimization of the cost function

$$J = \sum_{i=N_1}^{N_1+N} [w(k+i) - y(k+i)]^2 + \sum_{i=1}^{N_u} \lambda(i)[\Delta u(k+i-1)]^2, \qquad (12.6.5a)$$

where

N_1 = start of the output horizon,
N = length of the output horizon,
N_u = length of the control input horizon,
$\lambda(i)$ = is the positive control costing factor.

In the basic form, if $N_1 = 1$, $N_u = N$, and $\lambda(i) = \lambda$, using (12.6.4) the cost function (12.6.5a) can be restated as

$$J = \underline{\varepsilon}^T\underline{\varepsilon} + \lambda[\Delta u]^T[\Delta u]. \qquad (12.6.5b)$$

The control vector minimizing J is given by

$$\Delta \mathbf{u} = [G^TG + \lambda I]^{-1}G^T(\mathbf{w} - \hat{\mathbf{y}}_A). \qquad (12.6.6)$$

Hence the optimal control law,

$$u(k) = u(k-1) + \Delta u(k), \qquad (12.6.7)$$

$\Delta u(k)$ being the first element of $\Delta \mathbf{u}(k)$ in (12.6.6).

Example 12.6 Compute the generalized predictive control law for the process given by

$$(1-3q^{-1}+2q^{-2})y(k) = (0.5-0.8q^{-1})u(k-1) + \frac{e(k)}{\Delta},$$

minimizing (12.6.5a), assuming $\lambda = 0.5$, $N_1 = 1$, $N_u = N = 3$, and the desired set point being known.

Here,

$$\Delta A(q^{-1}) = (1-q^{-1})(1-3q^{-1}+2q^{-2}) = 1-4q^{-1}+5q^{-2}+2q^{-3}.$$

$$B(q^{-1}) = 0.5 - 0.8q^{-1}, \text{ and } d = 1.$$

The parameters of $E_p(q^{-1})$ and $F_p(q^{-1})$ for $p = 1$ to 3 are computed using (12.6.2):

$$1 = E_p(q^{-1})\Delta A(q^{-1}) + q^{-p}F(q^{-1}).$$

For $p = 1$,

$$E_p(q^{-1}) = 1, \text{ and } F_p(q^{-1}) = f_{o1}+f_{11}q^{-1}+f_{21}q^{-2};$$

so

$$1 = (1-4q^{-1}+5q^{-2}-2q^{-3}) + q^{-1}(f_{o1}+f_{11}q^{-1}+f_{21}q^{-2}).$$

Hence $f_{01} = 4$, $f_{11} = -5$, $f_{12} = 2$.

For $p = 2$,

$$E_p(q^{-1}) = 1+e_1q^{-1}, \text{ and } F_p(q^{-1}) = f_{o2}+f_{12}q^{-1}+f_{22}q^{-2};$$

So

$$1 = (1+e_1q^{-1})(1-4q^{-1}+5q^{-2}-2q^{-3}) + q^{-2}(f_{o2}+f_{12}q^{-1}+f_{22}q^{-2}).$$

Hence $\quad e_1 = 4,\ f_{02} = 11,\ f_{12} = -18,\ f_{22} = 8.$

For p = 3,

$$E_p(q^{-1}) = 1+e_1q^{-1}+e_2q^{-2}, \text{ and } F_p(q^{-1}) = f_{o3}+f_{13}q^{-1}+f_{23}q^{-2};$$

So,

$$1 = (1-e_1q^{-1}+e_2q^{-2})(1-4q^{-1}+5q^{-2}-2q^{-3}) + q^{-3}(f_{o3}+f_{13}q^{-1}+f_{23}q^{-2}).$$

Hence, $\quad e_1 = 4,\ e_2 = 11,\ f_{03} = 26,\ f_{13} = -47,\ f_{23} = 22.$

Since $\quad G_p(q^{-1}) = E_p(q^{-1})B(q^{-1}),$

$$\begin{aligned} G_1(q^{-1}) &= 0.5-0.8q^{-1}, \\ G_2(q^{-1}) &= (1+4q^{-1})(0.5-0.8q^{-1}) = (0.5+1.2q^{-1}-3.2q^{-1}), \\ G_3(q^{-1}) &= (1+4q^{-1}+11q^{-2})(0.5-0.8q^{-1}) \\ &= 0.5+1.2q^{-1}+2.3q^{-2}-8.8q^{-3}. \end{aligned}$$

Following (12.6.3),

$$\begin{aligned} \hat{y}(k+1|k) &= 4y(k) - 5y(k-1) + 2y(k-2) - 0.8\Delta u(k-1) \\ &\quad + 0.5\Delta u(k), \\ \hat{y}(k+2|k) &= 11y(k) - 18y(k-1) + 8y(k-2) - 3.2\Delta u(k-1) \\ &\quad + 1.2\Delta u(k) + 0.5\Delta u(k+1), \\ \hat{y}(k+3|k) &= 26y(k) - 47y(k-1) + 22y(k-2) - 8.8\Delta u(k-1) \\ &\quad + 2.3\Delta u(k) + 1.2\Delta u(k+1) + 0.5\Delta u(k+2). \end{aligned}$$

That is

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \hat{y}(k+3|k) \end{bmatrix} = \begin{bmatrix} 4 & -5 & 2 & -0.8 \\ 11 & -18 & 8 & -3.2 \\ 26 & -47 & 22 & -8.8 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \Delta u(k-1) \end{bmatrix} + \begin{bmatrix} 0.5 & & \\ 1.2 & 0.5 & \\ 2.3 & 1.2 & 0.5 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \end{bmatrix}$$

$$= \mathbf{y_A} + \mathbf{G\Delta u}.$$

Thus $\mathbf{y_A}$ and $\mathbf{G}$ being known, using known values for $\mathbf{w}$ and λ, $\mathbf{\Delta u}$ given by (12.6.6) can be computed, of which $\Delta u(k)$ is of interest (see Sec.12.8 for discussions on implementation aspects).

12.7 LRPC: DESIGN CONSIDERATIONS

The pulse response model based LRPC, the step response model based LRPC and GPC have been shown to have the same generic structure as regards the cost-function minimized and the control law generated. The difference in the performance of these LRPC schemes is due to (a) the different process models used, and (b) the different cost-functions optimized.

The four factors which characterize the LRPCs with respect to the cost-function (12.6.5a) are

(1) the starting horizon (N_1)
(2) the length of the horizon (N)
(3) the length of the control horizon (N_u) and
(4) the scalar cost on the control increments (λ).

From the process characteristics point of view, whether the controller can handle nonminimum-phase processes, open-loop unstable processes or variable time-delay processes, is of interest, and is discussed in this section. A nonminimum-phase process has a G which is rank deficient. Correct choice of control-cost λ and time-delay d is usually difficult, and it is important that the controller performs well even with crude choices, typically with $\lambda = 0$ and $d = 1$.

Starting horizon

Ideally the starting horizon N_1 should be equal to the maximum time-delay between the input and output. For all three models (12.4.1), (12.5.1) and (12.6.1), a default value of $N_1 = 1$ has been used. The time-delay should preferably be overestimated. An underestimated time-delay (i.e. $N_1 > d$) leads to a nonminimum-phase representation of the process even though the process may be actually minimum phase. Algebraically, if $N_1 < d$, first $d-N_1$ rows of matrix G in (12.3.3) can be zero, so G^TG will be noninvertible, and a nonzero λ will be required to ensure invertibility of $[G^TG + \lambda I]$. Alternatively, a method which can be applied to invert even rank-deficient matrices has to be used (see for example SVD based implementation, discussed in Sec.12.8).

If specifying the time-delay d is difficult, it may be assumed that $d = N_1 = 1$, and the LRPCs will still be able to produce satisfactory control.

Length of the horizon

One of the main features of LRPCs is that a horizon N (>1) is considered over which the predicted output should agree with the set point, whereas in the case of STC, N = 1. For a nonminimum-phase process, the step response of which has an initial negative going response (for example see Fig. 12A.2 in Appendix 12A), the horizon should be long enough to include the positive-going response; otherwise the control will not be stable. In general, the horizon should preferably be long enough to correspond to the rise time of the process.

Length of the control horizon and the scalar cost

The length of the control horizon N_u defines the length of the vector Δu in (12.3.3). For a finite N_u, an infinite scalar cost λ is assumed for the control increments beyond the control horizon N_u; in other words,

$$\Delta u(k+j-1) = 0, \qquad\qquad j > N_u.$$

That is, beyond N_u, the control is assumed to remain unchanged at $u(k+N_u)$ value.

Remark: For LRPCs based on pulse response models, since control increments with respect to $u(k-1)$ are considered, for a finite N_u, $\delta u(k+j-1) = 0$, $(j>N_u)$ and hence the control remains unchanged at $u(k-1)$ value from $(k+N_u)$ onwards. □□

Consider, the different possible values for N_u as follows.

(a) If $N_u = 1$, $\Delta u(k+i) = 0$ for $i \geq 1$. At $(k+1)$, $(k+2)$,..., etc., since the control remains unchanged at $u(k)$, the process may be considered to be running open-loop. Now, if $\lambda = 0$, the control will fail for an open-loop unstable process. However, open-loop stable, nonminimum-phase processes can be controlled with $N_u = 1$ and $\lambda = 0$, if the horizon N is sufficiently long as discussed earlier. The main attraction is that for $N_u = 1$. **G** is an N-vector, and the control computation (12.3.5) becomes scalar computation.

(b) If $N_u = N$, **G** is an NxN matrix; **G** has to be a full rank matrix, for $[\mathbf{G}^T\mathbf{G} + \lambda \mathbf{I}]^{-1}$ to exist with $\lambda = 0$, otherwise $[\mathbf{G}^T\mathbf{G} + \lambda \mathbf{I}]^{-1}$ has to be computed using SVD.

(c) If $1<N_u<N$, **G** will be an NxN_u rectangular matrix. N_u can be chosen equal to the rank of **G**, to ensure invertibility of $\mathbf{G}^T\mathbf{G}$. For example, if N = 4 and $N_u = 3$,

$$G = \begin{bmatrix} g_o & & 0 \\ g_1 & g_o & \\ g_2 & g_1 & g_o \\ g_3 & g_2 & g_1 \end{bmatrix};$$

even if $g_o = 0$, G will be full rank and G^TG will be invertible. Thus N_u should be at least equal to the number of unstable or nearly unstable or oscillatory poles of the controller, for stable control performance even with $\lambda = 0$. From a computation point of view, small N_u is preferable.

Remarks

(a) *Usual settings*

The usual setting is N_1 = maximum possible time delay and $N_u = 1$. N should be at least equal to the model order and cover the rise time of the process; typically $N = 10$. Complex processes require $N_u>1$.

(b) *Process models*

(i) The required model order for CARIMA representation is much less than that for a pulse response model or a step response model, which is the main advantage of using GPC.

(ii) All the LRPC schemes discussed in this chapter have integrating noise structure, which ensures disturbance rejection with zero offset.

(c) *Alternative cost-functions*

Since LRPC involves optimization of a finite-horizon cost-function (as against infinite stage cost minimization in the case of the LQ control discussed in the next chapter), constrained cost-functions or even nonquadratic cost-functions may be accommodated.

(d) *Comparative features of IDCOM, MAC and DMC methods*

The conventional IDCOM and MAC control strategies which are pulse response model based, and use $N_1 = 1$, $N_u = N$ and $\lambda = 0$. Since G^TG has to be invertible for stable control, nonminimum-phase processes or processes with varying time delay cannot be controlled. The conventional DMC, which is step response model based, use $N_1 = 1$, whereas N, N_u, λ can be chosen as desired. So with proper design, nonminimum-phase processes or processes with varying time-delay can be controlled.

12.8 IMPLEMENTATION ASPECTS OF LRPC

The parameter estimation aspects have been discussed in Chapter 3. The computation of the predictive control law involves two main problems:

(i) multistep prediction of the output over the horizon of interest, and

(ii) computation of the matrix inverse of the generic form of $[G^TG + \lambda I]^{-1}$.

Multistep prediction of output has been discussed in Chapter 5. The present discussions are confined to the latter matrix inversion problem. Two popular methods are discussed.

Implementation using singular value decomposition

Matrix inversions, using the singular value decomposition (SVD), studied in Sec.3.3.2, will be followed here.

Consider the SVD of the mxn matrix G: $G = U\Sigma V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix: $\Sigma = \text{diag}\ [\sigma_1\ \sigma_2\ \dots\ \sigma_p\ :0]$, where the singular values σ_1 etc. appear along the diagonal in nonincreasing order: $\sigma_1 \ge \sigma_2 \ge \dots \sigma_p \ge 0$. Hence

$$\begin{aligned} G^TG + \lambda I &= V\Sigma U^TU\Sigma V^T + V\lambda IV^T \\ &= V[\Sigma^2 + \lambda I]V^T. \end{aligned} \qquad (12.8.1)$$

So

$$[G^TG + \lambda I]^{-1} = V[\Sigma^2 + \lambda I]^{-1}V^T,$$

leading to

$$u_c = [G^TG + \lambda I]^{-1}G^T[w-y_A] = V[[\Sigma^2 + \lambda I]^{-1}\Sigma]U^T[w-y_A]. \qquad (12.8.2)$$

Here, $[\Sigma^2 + \lambda I]^{-1}\Sigma$ is a diagonal matrix of which

$$\text{the i-th diagonal element} = \frac{\sigma_i}{\sigma_i^2 + \lambda}.$$

The singular values which are insignificantly small may be eliminated (through truncation of the diagonal matrix $[(\Sigma^2 + \lambda I)^{-1}\Sigma]$) before computing the control law in (12.8.2). Since the first element in u_c is to be calculated to determine u(k), only the first row of V will be used in the computation using (12.8.2).

Implementation using UD factorization

UD factorization of a square matrix **P** is given by: $\mathbf{P} = \mathbf{UDU}^T$, **U** being an upper triangular and **D** being a diagonal matrix. The present matrix inversion problem has some structural similarity with the recursive least squares estimation problem for which UD factorization is popularly used (see Sec.3.4.2 and Appendix 3A). **U** should not be confused with the orthogonal matrix **U** obtained through SVD, which is different.

Consider an upper triangular matrix **G**:

$$\mathbf{G} = \begin{bmatrix} g_o & & 0 \\ g_1 & g_o & \\ g_2 & g_1 & g_o \\ g_3 & g_2 & g_1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \\ \mathbf{r}_4^T \end{bmatrix} \text{ (say).}$$

Hence

$$[\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}]^{-1} = \left(\sum_{i=1}^{4} \mathbf{r}_i\mathbf{r}_i^T + \lambda\mathbf{I}\right)^{-1}. \tag{12.8.3}$$

Referring to the parameter estimation problem, the covariance update is given by

$$\mathbf{P}^{-1}(k) = \mathbf{P}^{-1}(k-1) + \mathbf{x}(k)\mathbf{x}^T(k), \tag{12.8.4}$$

where **x** is the new data (column) vector received. Using the similarity between (12.8.3) and (12.8.4), and defining $\mathbf{P}^{-1} = [\mathbf{G}^T\mathbf{G}+\lambda\mathbf{I}]$, (12.8.3) can be computed recursively as follows:

(i) Initialize $\mathbf{P}^{-1}(k-1) = \lambda\mathbf{I}$, that is

$\mathbf{P}(k-1) = \mathbf{I}/\lambda = \mathbf{U}(k-1)\mathbf{D}(k-1)\mathbf{U}^T(k-1)$.

So $\mathbf{U}(k-1) = \mathbf{I}$, and $\mathbf{D}(k-1) = \mathbf{I}/\lambda$.

Let $j = 1$.

(ii) Define $\mathbf{x}(k) = \mathbf{r}_j$

(iii) Use UD measurement update routine to compute U(k) and D(k) where

$\mathbf{P}(k) = \mathbf{U}(k)\mathbf{D}(k)\mathbf{U}^T(k)$.

(iv) Increment j and go to (ii) until the computation of (12.8.3) is complete, which is given by the final **P**. If **G** is an mxn matrix (m>n), m calls to the UD measurement routine will be necessary.

Remarks
(a) UD factorization based implementation can be computed recursively, and is computationally more efficient than the SVD based approach, which cannot be computed recursively.
(b) SVD based implementation is numerically more robust. The noninvertibility of **G**, with or without nonzero λ, can be easily handled with appropriate truncation of **U**, Σ, and **V** matrices. The rank of **G** obtained through the singular values can also be used as a diagnostic feature; the rank deficiency of **G** indicates the number of unstable zeros in the process and the desired length of the control horizon.

12.9 CONCLUSIONS

A brief outline of some of the popular predictive control methods has been presented. All the methods studied have been reported to have been successful in real-life applications.

The self-tuning control (STC) and the long range predictive control (LRPC) methods basically involve two stages: parameter estimation and design of the control law; in both these stages different choices are possible, which ascribe different properties to the control algorithms. In STC, the costing (λ) of the control increments protects the controller from instability in nonminimum-phase processes, although the optimum choice is not easy. The two prime stabilizing features of LRPC methods are the costing of the future control increments and optimization of a multistage cost-function over a predictive horizon. These controllers can handle nonminimum-phase processes with relative ease (even with vanishingly small λ), and in general possess better stability and robustness properties than STC. Although the controller design part is on stronger grounds, more work needs to be done to make the parameter estimator equally robust.

Out of the different available designs of LRPCs, it is the generalized predictive control (GPC) strategy, which is of particular interest, as most other LRPC designs can be shown to be subsets of GPC.

For successful implementation, it is usually necessary to install safety constraints on the estimated parameter values and also on the control increments based on the prior knowledge of the system. The inherent nonlinearity of the associated equipments (like the control valve etc.) also need to be taken into consideration.

REFERENCES

Remarks: The idea of self-tuning in adaptive control appears in the classic paper [17] and is revived in [21]; the two key papers introducing self-tuning ideas to control are [1,7], further extension is followed by many papers, e.g., [8]; an early reference for multivariable design is [3]. Control of nonminimum-phase systems is studied in [5]. Self-tuning ideas (including pole placement control) and their applications are discussed in [26]. The properties of STC are discussed in the book [16]. Application of adaptive prediction and control as operator-guide in metal industries is reported in [13,18], the latter concerning Sec.12.2.3.

Different versions of long range predictive control (LRPC) methods are reviewed in [15]. Impulse response based designs are studied in [22,23,4], and the step response based designs are treated in [12,19]; these algorithms are compared and improved in [11]. GPC is discussed in [9,10]; further details of the algorithm and the implementation aspects are given in [20]. Predictive controllers discussed in [14,27] are subsets of GPC. The books [6,25] are devoted to predictive control. Adaptive control is broadly covered in [2].

[1] Åström, K.J., and B. Wittenmark (1973): 'On self-tuning regulators', *Automatica*, 9, 185-199.

[2] Åström, K.J., and B. Wittenmark (1989): *Adaptive Control*, Addison-Wesley, Reading, Mass.

[3] Borisson, U. (1979): 'Self-tuning regulators for a class of multivariable systems', *Automatica*, 15, 209-215.

[4] Bruijn, P.M., and H.B. Verbruggen (1984): 'Model algorithmic control using impulse response models', *Journal A*, 25, 69-74.

[5] Clarke, D.W. (1984): 'Self-tuning control of non-minimum-phase systems', *Automatica*, 20, 501-517.

[6] Clarke, D.W. (Ed.) (1994): *Advances in Model-based Predictive Control*, Oxford Univ. Press, Oxford.

[7] Clarke, D.W., and P.J. Gawthrop (1975): 'Self-tuning controller', *Proc. IEE*, 122, 929-934.

[8] Clarke, D.W., and P.J. Gawthrop (1979): 'Self-tuning control', *Proc. IEE*, 126, 633-640.

[9] Clarke, D.W., C.H. Mohtadi, and P.S. Tuffs (1987a): 'Generalized predictive algorithm - Part I, the basic algorithm', *Automatica*, 23, 137-148.

[10] Clarke, D.W., C.H. Mohtadi, and P.S. Tuffs (1987b): 'Generalized predictive algorithm - Part II, extensions and interpretations', *Automatica*, 23, 149-160.

[11] Clarke, D.W., and L. Zhang (1987): 'Long-range predictive control using weighting-sequence model', *Proc. IEE*, 134, Pt.D, 187-195.

[12] Cutler, C.R., and B.L. Ramaker (1980): 'Dynamic Matrix Control , A Computer Control Algorithm', *JACC*, San Francisco.

[13] De Keyser, R.M.C., and A. Van Cauwenberghe (1981): 'A self-tuning multistep predictor application', *Automatica*, 17(1), 167-174.

[14] Dekeyser, R.M.C., and A. Van Cauwenberghe (1985): 'Extended Prediction Self-adaptive Control', *IFAC Symp. on Identification and System Parameter Estimation*, New York, 1255-1260.

[15] Garcia, C.E., D.M. Prett, and M. Morari (1989): 'Model predictive Control Theory and Practice - a Survey', *Automatica*, 25, 3, 335-348.

[16] Goodwin, G.C., and K.S. Sin (1984): *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, N.J.

[17] Kalman, R.E., (1958): 'Design of a self-optimizing control system', *Trans. ASME*, 80, 468-478.

[18] Kanjilal, P.P., and E. Rose (1986): 'Application of adaptive prediction and control methods for improved operations of the sintering process', *Ironmaking and Steelmaking*, 13(6), 289-293.

[19] Martin, G.D., J.M. Caldwell, and T.E. Ayral (1986): 'Predictive control applications for the petroleum refining industry', *Japan Petroleum Institute Petroleum Refining Conference*, Tokyo.

[20] Mohtadi, C.H. (1987): 'Advanced Self-tuning Algorithms', D.Phil. Thesis, OUEL 1689/87, Dept. of Eng.

Science, University of Oxford.

[21] Peterka, V. (1970): 'Adaptive digital regulation of noisy systems', *Preprints 2nd IFAC Symposium on Identification and Process Parameter Estimation*, Prague.

[22] Richalet, J., A. Rault, J.L. Tesud, and J. Papon (1978): 'Model predictive heuristic control: applications to industrial processes', *Automatica*, 14, 413–428.

[23] Richalet, J. (1993): 'Industrial applications of model based predictive control', *Automatica*, 29(5), 1251–1274.

[24] Rouhani, R., and R.K. Mehra (1982): 'Model algorithmic control (MAC), basic theoretical properties', *Automatica*, 18(3), 401–414.

[25] Soeterboek, R. (1992): *Predictive Control: A Unified Approach*, Prentice Hall, New York.

[26] Wellstead, P.E., and M.B. Zarrop (1991): *Self-tuning Systems: Control and Signal Processing*, Wiley, Chichester

[27] Ydstie, B.E. (1984): 'Extended Horizon Adaptive Control', Paper 14.4/E.4, *9th IFAC World Congress*, Budapest.

CHAPTER 13

PREDICTIVE CONTROL (Part-II): STATE-SPACE MODEL BASED

Predictive controllers can be designed in the state-space framework with specified end-point reference conditions.

13.1 INTRODUCTION

Long range predictive control (LRPC) methods formulated using transfer function models were discussed in Chapter 12. This chapter is devoted to the study of state-space formulation of linear quadratic (LQ) controllers, which form another popular class of predictive controllers. Here, the process is represented by a *linear* state-space model, and the cost criterion is a *quadratic* function of the states and the control inputs. If the disturbances to the process (as expressed by the model), are Gaussian in nature, the LQ control is referred to as the linear quadratic Gaussian (LQG) control.

The following comparative features of state-space LQ control and LRPC may be noted.

(a) Both the classes of control consider a linear model of the process, and both minimize scalar quadratic cost-functions to compute the control law u(k).

(b) The LRPC methods use the present time k as the terminal reference point, and the control u(k) is determined based on multistep predictions of the output over a finite horizon in the forward direction into the future. On the other hand, the state-space based LQ control considers a time-point at the end of a horizon (k+N) as the terminal point; the control u(k) can be computed through backward recursion of the control algorithm to the present time, satisfying the specified terminal conditions. N, the length of the horizon, can be finite or infinite.

(c) Under certain conditions, the LQ control in state-space formulation and the LRPC produce identical results.

Discussions on state-space formulation of LQ control is widely covered in many papers and texts; this chapter provides a brief exposure to the basic concepts, results and the implementation aspects. There are two basic issues in the design of LQ controllers: (i) whether the process is modelled as a deterministic or a stochastic process, and (ii) whether the parameters of the process model are time invariant and known or are time varying and unknown. The present study covers the design of LQ control both for deterministic and stochastic processes; both time invariant and time varying processes are considered.

The study of LQ control of a time invariant deterministic process is presented in Sec.13.2. The control law u(k) has two elements: $u(k) = -\mathbf{k}^T(k)\mathbf{x}(k)$, where $\mathbf{k}$ is the *control gain* and $\mathbf{x}$ is the *estimated state*. In the case of a stochastic process, the LQ controller may be designed by invoking the separation theorem, which permits the state estimation and the controller design to be considered as two separate problems. The separation theorem is introduced in Sec.13.3, and the design of the stochastic LQ controller as an optimal state estimator associated with an optimal deterministic controller is discussed. Sec.13.4 demonstrates that the algebraic steps for the solution of the deterministic LQ control problem and the state estimation problem are dual of each other.

For time varying stochastic processes, the process parameters are estimated from an input-output model, and the estimated parameters are used for state estimation, which is discussed in Sec.13.5. The computation of the deterministic control law and the associated implementation aspects are considered in Sec.13.6.

This chapter is supported by three appendices. Derivation of the LQ control law for a multivariable process is detailed in Appendix 13A (note that the studies presented in this chapter consider a single-input single-output process). Appendix 13B discusses the transmittance matrix formulation and its implementation, which concern the estimation of the state in a stochastic environment. Appendix 13C presents the UD time-update algorithm and its implementation which is used in the computation of the control law.

13.2 LQ CONTROL OF A DETERMINISTIC PROCESS

Consider a single-input (u) and a single-output (y) process represented by the deterministic model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \tag{13.2.1}$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k), \tag{13.2.2}$$

where **x** is an n-state-vector, y is the measured output and u is the deterministic control input; **A** is an nxn matrix, and **b** and **c** are n-vectors. The initial state **x**(0) is assumed to be known. Since the model is deterministic, it is understood that there is no uncertainty (in terms of process noise or measurement noise etc.) present, and complete knowledge of the states is available.

The objective is to produce the optimal control decisions u(k), u(k+1),..., u(k+N−1), so as to minimize the scalar cost function

$$J = \mathbf{x}^T(k+N)\mathbf{Q}_N\mathbf{x}(k+N) + \sum_{i=k}^{k+N-1}(\mathbf{x}^T(i)\mathbf{Q}\mathbf{x}(i) + \lambda u^2(i)), \tag{13.2.3}$$

where $\mathbf{Q}_N$ and **Q** are symmetric positive semidefinite matrices, and λ is a positive scalar cost; λ may be conditionally zero or vanishingly small as discussed later.

There are various methods for solving the present optimal control problem (Strejc, 1981). One of the popular approaches is the method of dynamic programming, which is based on the principle of optimality.

Remark: The *principle of optimality*, (Bellman, 1957) states that an optimal control sequence has the property that whatever the initial state and the initial control are, the remaining controls must constitute an optimal sequence with regard to the state resulting from the initial control.

□□

The detailed solution of the optimal control problem using dynamic programming appears in Appendix 13A; the main results are presented here.

Let the cost at the last stage of the horizon {k,k+N} be defined as

$$J_N = \mathbf{x}^T(k+N)\mathbf{P}(k+N)\mathbf{x}(k+N),$$

where $\mathbf{P}(k+N) = \mathbf{Q}_N$. It can be shown that starting from the prespecified terminal condition $\mathbf{P}(k+N) = \mathbf{Q}_N$, the optimal control law can be computed through backward recursion from one stage to the next, with each stage having identical structure, until the present stage is reached. At each stage

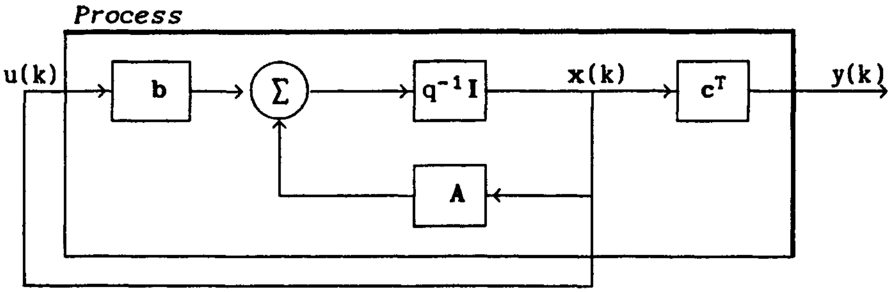


Figure 13.2.1 Deterministic state-space model of the process with optimal LQ control. **k(k)** is the Kalman gain or the feedback gain.

i, (k≤i<k+N) the gain vector **k**(i) is computed using **P**(i+1), and the control u(k) is determined; **P**(i) is next computed to be used in the next stage i-1 until i=k.

Summing up, the general solution of the deterministic optimal control problem is given by

$$\mathbf{k}^T(k) = (\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(k+1)\mathbf{A}, \qquad (13.2.4)$$

$$\mathbf{P}^*(k) = \mathbf{P}(k+1) - \mathbf{P}(k+1)\mathbf{b}(\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(k+1), \qquad (13.2.5)$$

$$\mathbf{P}(k) = \mathbf{Q} + \mathbf{A}\mathbf{P}^*(k)\mathbf{A}, \qquad (13.2.6)$$

$$u(k) = -\,\mathbf{k}^T(k)\mathbf{x}(k), \qquad (13.2.7)$$

where **P**(k+N) = $\mathbf{Q}_N$. The schematic diagram of the deterministic process with LQ control u is shown in Fig.13.2.1.

Remarks

(1) λ=0 may be permitted in (13.2.3) only if $\mathbf{b}^T\mathbf{P}(k+1)\mathbf{b}$ in (13.2.4) is ensured to be nonzero, however if the implementation involves explicit inversion of λ, a positive λ will be required.

(2) The first term in (13.2.3) penalizes the terminal deviation of the state from zero set-point. This is why the present control problem is called a linear *regulator* problem. Non-zero set-points can be accommodated by expressing

the measurement equation in terms of $\varepsilon(k)$ instead of $y(k)$:

$$\varepsilon(k) = y(k) - w(k) = c^T x(k),$$

w being the set-point; for further details see Sec.13.5.

13.3 SEPARATION THEOREM AND CONTROL OF A STOCHASTIC PROCESS

In the last section, optimal control law was derived for a linear deterministic model; the process was assumed to be free from unknown disturbances, and complete knowledge of the states was assumed to be available. In practice, the process may not be exactly linear, nor are the states likely to be exactly known; there may be noise associated with the measurements, and unknown disturbances may be acting on the process. Even if the process is time invariant, due to the various uncertainties present, the variables **x** and y which characterize the process become stochastic variables, and the consequent model is called a stochastic model.

13.3.1 The Control Problem

Process model

Let the process be represented by the model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k) + \mathbf{v}_1(k), \tag{13.3.1}$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k) + v_2(k), \tag{13.3.2}$$

where the process noise $\{\mathbf{v}_1(k)\}$ and the measurement noise $\{v_2(k)\}$ are assumed to be zero-mean Gaussian white sequences with known statistics. The initial state $\mathbf{x}(0)$ is a zero-mean Gaussian random vector with positive semidefinite covariance matrix P(0). Usually $\mathbf{x}(0)$, $\{\mathbf{v}_1(k)\}$ and $\{v_2(k)\}$ are assumed to be mutually independent, which is not a limitation.

Control objective

The objective is to compute the control sequence u(k), u(k+1),...,u(k+N-1) so as to minimize the scalar cost

$$J = E\{\mathbf{x}^T(k+N)Q_N\mathbf{x}(k+N) + \sum_{i=k}^{k+N-1}(\mathbf{x}^T(i)Q\mathbf{x}(i) + \lambda u^2(i))\}, \tag{13.3.3}$$

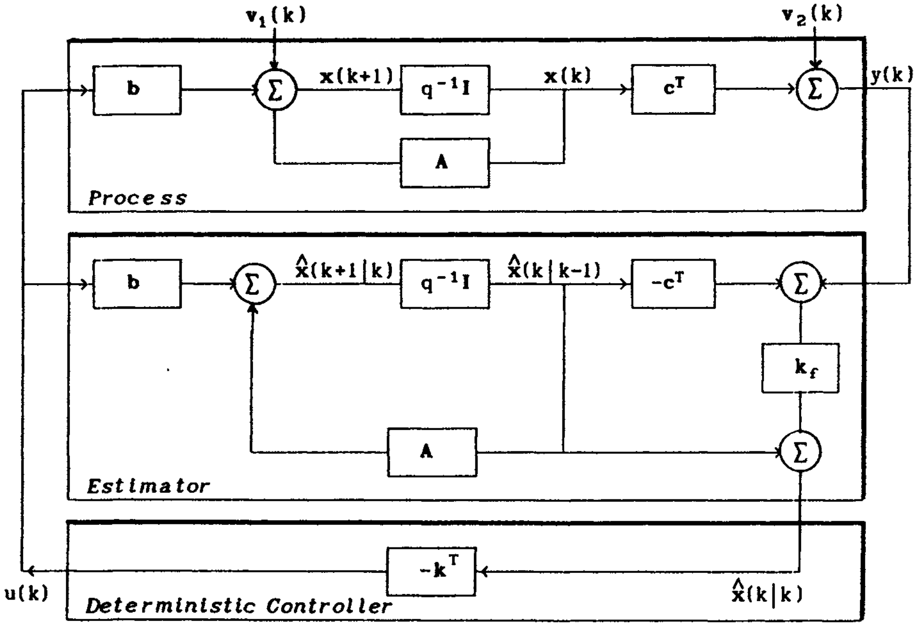


Figure 13.3.1 LQ control of a stochastic process using the separation theorem. The state estimator and the deterministic controller are treated as separate problems.

where Q_N and Q are positive semidefinite symmetric matrices, and λ is a positive scalar constant.

13.3.2 Separation Theorem and Controller Synthesis

According to the separation theorem, the present optimization problem (concerning the determination of the optimal control sequences for the stochastic process) has two *separate* components, which can be individually optimised (see Fig.13.3.1):

(i) the deterministic optimal control problem, which is solved for the optimal LQ controller gain $\mathbf{k}(k)$ (13.2.4), and

(ii) the stochastic state estimation problem which is solved for the optimal state estimate $\hat{\mathbf{x}}(k|k)$.

The desired optimal solution is given by

$$u(k) = -\mathbf{k}^T(k)\hat{\mathbf{x}}(k|k).$$

Features consequent to separation theorem

(a) The LQ control (feedback)) gain **k**(k) is independent of the statistical characteristics of the process model (13.3.1 - 13.3.2), i.e. **P**(0) and the noise statistics. **k**(k) is also independent of the observed data y(k).

(b) The optimal state ($\hat{\mathbf{x}}(k|k)$) estimation is independent of the optimal control problem, and hence independent of the weighting matrices $\mathbf{Q}_N$, **Q** and the scalar λ in the cost function (13.3.3). The past control inputs u(k-1), u(k-2) etc. are considered deterministic data. The state may be estimated using the Kalman filter, discussed in Sec.6.6.

(c) Computation of **k**(k) involves backward iteration from the reference terminal point, whereas computation of $\hat{\mathbf{x}}(k|k)$ involves forward iteration starting from the initial state.

13.4 DUALITY BETWEEN LQ CONTROL AND STATE ESTIMATOR

The optimal deterministic LQ control system and the optimal state estimator are structurally identical (Kalman, 1960); this relationship can be shown as follows.

State estimation by Kalman filter

Let a single-input single-output process be expressed as

$$\mathbf{x}(k+1) = \bar{\mathbf{A}}\mathbf{x}(k) + \bar{\mathbf{b}}u(k) + \mathbf{v}_1(k), \qquad (13.4.1a)$$

$$y(k) = \bar{\mathbf{c}}^T\mathbf{x}(k) + v_2(k); \qquad (13.4.2a)$$

the noise sequences $\{\mathbf{v}_1(k)\}$ and $\{v_2(k)\}$ are assumed to be zero-mean, Gaussian white, with the covariances:

$$E\{\mathbf{v}_1(k)\mathbf{v}_1^T(k)\} = \mathbf{R}_1, \qquad E\{v_2^2(k)\} = R_2,$$

and the initial state **x**(0) is assumed to be a zero-mean Gaussian random vector, with positive semidefinite covariance matrix $\bar{\mathbf{P}}(0)$.

The Kalman filter estimate of the state is given by

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{k}_f(y(k) - \bar{\mathbf{c}}^T\hat{\mathbf{x}}(k|k-1)), \qquad (13.4.3a)$$

where the optimal filter gain $\mathbf{k}_f(k)$ is obtained from the recursive solution of the Riccati equation as

$$\mathbf{k}_f(k) = \bar{\mathbf{P}}(k|k-1)\bar{\mathbf{c}}(\bar{\mathbf{c}}^T\bar{\mathbf{P}}(k|k-1)\bar{\mathbf{c}} + R_2)^{-1}, \qquad (13.4.4a)$$

$$\bar{\mathbf{P}}^*(k|k) = \bar{\mathbf{P}}(k|k-1) - \bar{\mathbf{P}}(k|k-1)\bar{\mathbf{c}}(\bar{\mathbf{c}}^T\bar{\mathbf{P}}(k|k-1)\bar{\mathbf{c}}+R_2)^{-1}\bar{\mathbf{c}}^T\bar{\mathbf{P}}(k|k-1), \qquad (13.4.5a)$$

$$\bar{\mathbf{P}}(k+1|k) = \bar{\mathbf{A}}\bar{\mathbf{P}}^*(k|k)\bar{\mathbf{A}}^T + \mathbf{R}_1, \qquad (13.4.6a)$$

starting with the specified initial conditions:

$$\hat{\mathbf{x}}(0|-1) = \mathbf{x}(0) \quad \text{and} \quad \mathbf{P}(0|-1) = \mathbf{R}_o.$$

LQ control of a deterministic process

Consider the deterministic model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \qquad (13.4.1b)$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k), \qquad (13.4.2b)$$

the optimal control minimizing the LQ cost function (13.2.3) is given by

$$u(k) = -\mathbf{k}^T(k)\mathbf{x}(k), \qquad (13.4.3b)$$

where the Kalman control gain **k(k)** is obtained from the recursive solution of the Riccati equation as follows:

$$\mathbf{k}^T(k) = (\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(k+1)\mathbf{A}, \qquad (13.4.4b)$$

$$\mathbf{P}^*(k) = \mathbf{P}(k+1) - \mathbf{P}(k+1)\mathbf{b}(\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(k+1), \qquad (13.4.5b)$$

$$\mathbf{P}(k) = \mathbf{Q} + \mathbf{A}\mathbf{P}^*(k)\mathbf{A}, \qquad (13.4.6b)$$

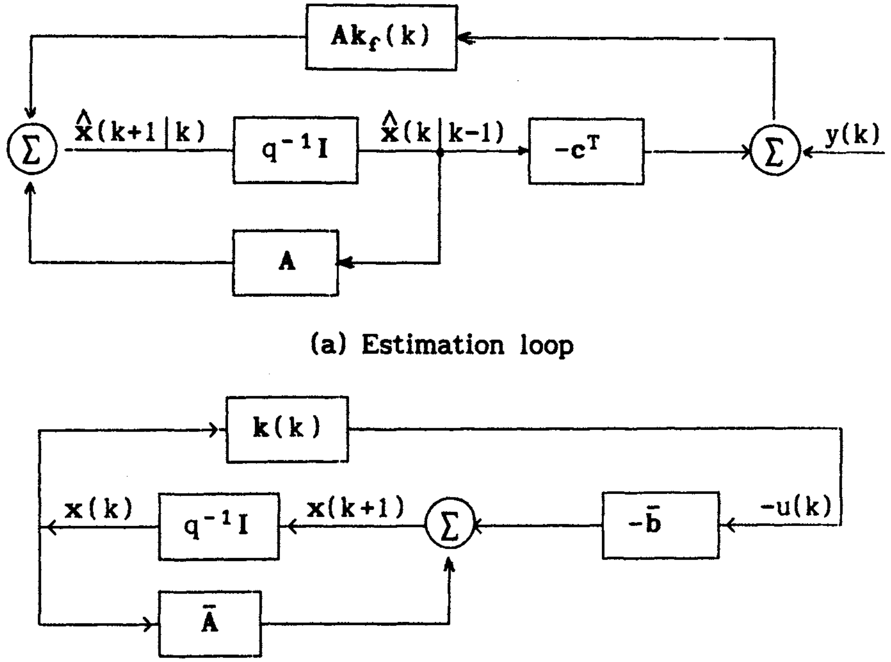
starting with the specified terminal condition $\mathbf{P}(k+N) = \mathbf{Q}_N$.

Remarks

(1) Note that the sets of Equations (13.4.4a - 13.4.6a) and (13.4.4b - 13.4.6b) are structurally dual of each other.

(2) In terms of implementation, the duality between the optimal filter and the LQ controller permits the use of the same measurement update routine for solving (13.4.4a, 13.4.5a) and (13.4.4b, 13.4.5b) for the filter and for the controller respectively. Similarly the same time-update routine can be used for solving equations (13.4.6a) and (13.4.6b) for the filter and the controller respectively. The measurement update routine can also be used for the recursive least squares estimation of the parameters in time varying processes, as discussed in the next section. The implementation aspects are treated in detail in Sec.13.7.

(3) The duality between the optimal state estimator and the



(a) Estimation loop

(b) Control loop

Figure 13.4.1 Duality between the estimation loop and the control loop. The duality is with respect to the determination of the filter gain $\mathbf{k}_f$ and the controller gain $\mathbf{k}$.

optimal deterministic LQ controller, shown in Fig.13.4.1, can be summarized (in terms of correspondence of the respective terms and features) as follows.

State estimator	*LQ controller*
$\bar{\mathbf{P}}(k\|k)$	$\mathbf{P}^*(k)$
$\bar{\mathbf{P}}(k+1\|k)$	$\mathbf{P}(k)$
$\bar{\mathbf{c}}$	$\mathbf{b}$
$\bar{\mathbf{A}}$	$\mathbf{A}^T$
R_2	λ
$\mathbf{R}_1$	$\mathbf{Q}$
$\mathbf{k}_f^T\bar{\mathbf{A}}$	$\mathbf{k}^T$
Backward iteration used to compute $\mathbf{P}(k)$ following the last stage: $\mathbf{P}(k+1)$.	Forward iteration used to compute $\mathbf{P}(k)$ following the last stage: $\mathbf{P}(k-1)$.

13.5 LQ CONTROL OF TIME VARYING PROCESSES

So far, the parameters of the state-space model have been assumed to be time invariant and known. In real-life most processes have time-varying unknown characteristics. So, with the progression of time, it is necessary to reestimate the parameters of the process model using the available measurements and information.

LQ control involves performing the following operations at every time step.

(1) Recursive estimation of the parameters of the input-output or transfer-function model.

(2) Estimation of the state $\hat{x}(k|k)$ using the estimated process parameters.

(3) Computation of the control law minimizing the specified cost function.

The LQ control problem is broadly specified in terms of the process model and the cost function to be minimized. In this section, the formulation of the LQ control problem in state-space format using the input-output data is presented, and the significance of the cost function minimized is also analysed. The problems of state estimation and the computation of control law are considered in the subsequent sections.

13.5.1 Process Models

CARMA model

Consider a single-input single-output process given by

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k), \qquad (13.5.1)$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_nq^{-n},$$
$$B(q^{-1}) = b_o + b_1q^{-1} + b_2q^{-2} + \dots + b_nq^{-n},$$
$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_nq^{-n};$$

d is the time delay between the input u and the output y, e represents the noise and the model uncertainties. $C(q^{-1})$ is assumed to be a stable polynomial (see Sec.2.4.2).

The equivalent state-space representation in observable canonical form is given by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k) + \mathbf{s}e(k), \qquad (13.5.2)$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k) + e(k), \qquad (13.5.3)$$

where

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 & & \mathbf{0} \\ -a_2 & 0 & 1 & & \\ \vdots & & & \ddots & \\ -a_n & & & & \ddots \\ 0 & & & & \ddots \\ \vdots & & & & 1 \\ 0 & & \mathbf{0} & & 0 \end{bmatrix},$$

$$\mathbf{b} = [0 \ \ldots \ 0 \ b_o \ b_1 \ \ldots \ b_n]^T, \qquad \mathbf{c} = [1 \ 0 \ \ldots \ 0]^T,$$

$$\mathbf{s} = [(c_1-a_1) \ (c_2-a_2) \ \ldots \ (c_n-a_n) \ 0 \ \ldots \ 0]^T. \qquad (13.5.4)$$

Note that (i) (d-1) leading elements of $\mathbf{b}$ are zeros, and (ii) the size of $\mathbf{x}$ = max (degree $A(q^{-1})$, degree $B(q^{-1})$ + d, degree $C(q^{-1})$).

CARIMA model

In case of the CARIMA representation of the process:

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k)/\Delta, \qquad (13.5.5)$$

that is

$$\tilde{A}(q^{-1})y(k) = B(q^{-1})\Delta u(k-d) + C(q^{-1})e(k), \qquad (13.5.6)$$

where

$$\tilde{A}(q^{-1}) = \Delta A(q^{-1}) = 1 + \tilde{a}_1 q^{-1} + \ldots + \tilde{a}_n q^{-n} + \tilde{a}_{n+1} q^{-(n+1)};$$

the corresponding state-space model is given by

$$\mathbf{x}(k+1) = \tilde{\mathbf{A}}\mathbf{x}(k) + \mathbf{b}\Delta u(k) + \tilde{\mathbf{s}}e(k), \qquad (13.5.7)$$

$$y(k) = \mathbf{c}^T\mathbf{x}(k) + e(k), \qquad (13.5.8)$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} -\tilde{a}_1 & 1 & 0 & & \mathbf{0} \\ -\tilde{a}_2 & 0 & 1 & & \\ \vdots & & & \ddots & \\ -\tilde{a}_{n+1} & & & & \ddots \\ 0 & & & & \ddots \\ \vdots & & & & 1 \\ 0 & & \mathbf{0} & & 0 \end{bmatrix},$$

$$\tilde{\mathbf{s}} = [(c_1-\tilde{a}_1) \ (c_2-\tilde{a}_2) \ldots (c_n-\tilde{a}_n) \ (-\tilde{a}_{n+1}) \ldots \ 0]^T, \qquad (13.5.9)$$

b and **c** being the same as in (13.5.4).

CARIMA model with nonzero set point

To incorporate nonzero set points (w(k)) in the N-stage minimization problem of the LQ controller design, it is necessary to know the future set points. If unknown, the future set points may be assumed to be the same as the present set point.

Introduce the output error

$$\varepsilon(k) = y(k) - w(k).$$

Hence

$$\tilde{A}(q^{-1})\varepsilon(k) = B(q^{-1})\Delta u(k-d) - A(q^{-1})\Delta w(k) + C(q^{-1})e(k). \qquad (13.5.10)$$

So the state-space model becomes

$$\mathbf{x}(k+1) = \tilde{\mathbf{A}}\mathbf{x}(k) + \mathbf{b}\Delta u(k) + \mathbf{w}\Delta w(k+1) + \tilde{\mathbf{s}}e(k), \qquad (13.5.11)$$

$$\varepsilon(k) = \mathbf{c}^T\mathbf{x}(k) + e(k), \qquad (13.5.12)$$

where

$$\mathbf{w} = [-1\ -a_1\ -a_2\ \ldots -a_n]^T,$$

$\tilde{\mathbf{A}}$, **b**, $\tilde{\mathbf{s}}$ and **c** being the same as in (13.5.7 - 13.5.8).

13.5.2 Cost Functions

For input-output models, considering a predictive horizon of length N, the LQ cost function is given by

$$J_1 = y^2(k+N) + \sum_{i=k}^{k+N-1}(y^2(i) + \lambda u^2(i)), \qquad (13.5.13a)$$

where the set-point is zero; in case of nonzero set-points (w), the equivalent cost function is given by

$$J_2 = (y(k+N)-w(k+N))^2 + \sum_{i=k}^{k+N-1}((y(i)-w(i))^2 + \lambda u^2(i)). \qquad (13.5.13b)$$

The problem with J_1 or J_2 is that, if the underlying process does not contain an integrator, a zero steady state offset between the output and the set point will require a nonzero control; this value of control may not correspond to the minimum cost (J_1 or J_2), even in the absence of disturbances.

The remedy is to consider a cost function where deviation of the control signal from a steady-state mean value is penalized, or instead of the actual control signal, increments of the control signal are penalized as follows:

$$J_3 = (y(k+N)-w(k+N))^2 + \sum_{i=k}^{k+N-1}((y(i)-w(i))^2+ \lambda(\Delta u(i))^2). \tag{13.5.14}$$

If the measurement equation is given by

$$y(k) = c^T x(k) + e(k)$$

as in (13.5.8), the cost functions J_1 may be approximated as

$$J_1' = E\{x^T(k+N)cc^T x(k+N) + \sum_{i=k}^{k+N-1}(x^T(i)cc^T x(i) + \lambda(u(i))^2)\}. \tag{13.5.15}$$

Similarly, if the measurement equation is given by

$$\varepsilon(k) = c^T x(k) + e(k),$$

as in (13.5.12), the cost function J_3 can be approximated to

$$J_3' = E\{x^T(k+N)cc^T x(k+N) + \sum_{i=k}^{k+N-1}(x^T(i)cc^T x(i)+\lambda(\Delta u(i))^2)\}. \tag{13.5.16}$$

With respect to the state-space representation (13.5.2 - 13.5.3) of the CARMA model, the LQ control law minimizing J_1' is given by

$$u(k) = -k^T(k)\hat{x}(k|k). \tag{13.5.17}$$

Similarly, with respect to the state-space representation (13.5.11 - 13.5.12) for the CARIMA model with nonzero set-point, the LQ control law minimizing J_3' is given by

$$\Delta u(k) = -k^T(k)\hat{x}(k|k); \tag{13.5.18}$$

the estimation of the state $\hat{x}(k|k)$ is discussed in Sec. 13.6, and computation of the gain $k(k)$ is discussed in Sec.13.7.

Remark: J_1' and J_3' will be equivalent to J_1 and J_3 respectively for deterministic cases, or if the sequence $\{e(k)\}$ is white.

13.6 ESTIMATION OF THE STATE $\hat{x}(k|k)$

The state can be estimated using the Kalman filter, (discussed in Sec.6.6). Alternatively, transmittance matrices may be used for estimation; this approach is followed in the present discussion.

The objective is to estimate the state $\hat{x}(k|k)$, given the process model (13.5.2 - 13.5.3) or (13.5.7 - 13.5.8). Note that in both cases, the process noise and the measurement noise are the same. In the estimation of $\hat{x}(k|k-1)$, the process noise can be eliminated in terms of the past input and the output signals. $\hat{x}(k|k-1)$ is measurement-updated to $\hat{x}(k|k)$ using a steady-state Kalman filter. Unlike the Kalman filter, the transmittance matrix approach does not require the knowledge of the covariances of the process noise and the measurement noise.

13.6.1 State Estimation from CARMA Model

The state-space representation for the CARMA model (13.5.1),

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})e(k),$$

can be interpreted to be in innovations form:

$$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + bu(k) + se(k), \qquad (13.6.1)$$

$$y(k) \quad = c^T\hat{x}(k|k-1) + e(k), \qquad (13.6.2)$$

where

$$\hat{y}(k|k-1) = c^T\hat{x}(k|k-1),$$

and hence

$$e(k) = y(k) - \hat{y}(k|k-1)$$

is referred to as the *innovations process*. Eliminating the noise e(k) from (13.6.1) using (13.6.2),

$$\hat{x}(k+1|k) = [A-sc^T]\hat{x}(k|k-1) + bu(k) + sy(k).$$

Let $[A-sc^T] = F$; so

$$\hat{x}(k|k-1) = [I-q^{-1}F]^{-1}[bu(k-1) + sy(k-1)], \qquad (13.6.3)$$

where following (13.5.4),

$$F = [A-sc^T] = \begin{bmatrix} -c_1 & 1 & 0 & & 0 \\ -c_2 & 0 & 1 & & \\ \vdots & & & & \\ -c_n & & & \ddots & \\ 0 & & & & \\ \vdots & & & & 1 \\ 0 & & 0 & & 0 \end{bmatrix}, \qquad (13.6.4)$$

which is a function of $C(q^{-1})$ alone. Since $C(q^{-1})$ is a stable polynomial, **F** is also a stable matrix (i.e. the eigenvalues of **F** will be within the unit circle).

Equation (13.6.3) is in a computationally inconvenient form as it involves polynomial matrix operations. A much simpler expression evolves from (13.6.3), if the time shift operations in $[\mathbf{I}-q^{-1}\mathbf{F}]^{-1}$ are absorbed in the corresponding input and output data vectors **u** and **y** respectively as follows. Let (13.6.3) be rewritten as

$$\hat{\mathbf{x}}(k|k-1) = \frac{1}{C(q^{-1})}\,[\mathbf{M}_u\mathbf{u}(k-1) + \mathbf{M}_y\mathbf{y}(k-1)], \qquad (13.6.5)$$

where

$$\mathbf{u}(k-1) = [u(k-1)\ u(k-2)\ \ldots\ ]^T,$$

$$\mathbf{y}(k-1) = [y(k-1)\ y(k-2)\ \ldots\ ]^T,$$

and $\mathbf{M}_u$ and $\mathbf{M}_y$, the transmittance matrices of **u**(k-1) and **y**(k-1) respectively, are given by the expressions

$$[\mathbf{I}-q^{-1}\mathbf{F}]^{-1}[\mathbf{b}u(k-1)] = \frac{1}{C(q^{-1})}\,[\mathbf{M}_u\mathbf{u}(k-1)],$$

and

$$[\mathbf{I}-q^{-1}\mathbf{F}]^{-1}[\mathbf{s}y(k-1)] = \frac{1}{C(q^{-1})}\,[\mathbf{M}_y\mathbf{y}(k-1)].$$

The transmittance matrices have simple implementation, as detailed in Appendix 13B.

Note that (13.6.5) is the estimate of **x** based on the available input and output data up to time k-1. At time k, using the additional information y(k), the estimate of the state may be measurement updated as

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{k}_f(y(k) - \mathbf{c}^T\hat{\mathbf{x}}(k|k-1)); \qquad (13.6.6)$$

$\mathbf{k}_f$ is the filter gain of the steady-state Kalman filter for the state estimator.

Again using (13.6.6), from (13.6.1 - 13.6.2),

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k) + \mathbf{b}u(k) + [\mathbf{s}-\mathbf{A}\mathbf{k}_f]e(k).$$

Thus the effect of noise in the progression of the state estimate is minimized for

$$\mathbf{s} = \mathbf{A}\mathbf{k}_f.$$

As **A** is singular, the solution of this equation is not unique. In the present case, one solution which is always valid (Lam, 1982a) is given by

$$\mathbf{k}_f = [1\ c_1\ c_2\ \ldots]^T.$$

Summing up, for a CARMA process,

$$\begin{aligned}\hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{k}_f(y(k) - \mathbf{c}^T\hat{\mathbf{x}}(k|k-1)) \\ &= \frac{1}{C(q^{-1})}[\mathbf{M}_u u(k-1) + \mathbf{M}_y y(k-1)] + \mathbf{k}_f e(k) \\ &= \frac{1}{C(q^{-1})}[\mathbf{M}_u u(k-1) + \mathbf{M}_y y(k-1) \\ &\qquad + \mathbf{k}_f(A(q^{-1})y(k) - B(q^{-1})u(k-d))].\end{aligned} \tag{13.6.7}$$

The LQ control law is given by

$$u(k) = -\mathbf{k}^T(k)\hat{\mathbf{x}}(k|k).$$

Remarks

(a) The optimal control law, $u(k) = -\mathbf{k}^T(k)\hat{\mathbf{x}}(k|k)$, should incorporate the knowledge of the measurement $y(k)$. If instead of $\hat{\mathbf{x}}(k|k)$, $\hat{\mathbf{x}}(k|k-1)$ is used, the control with *incomplete state information* will result, since the information $y(k)$ is left unutilized.

(b) If the noise observer polynomial $T(q^{-1})$ is used (as discussed in Sec.2.4.2), it simply replaces $C(q^{-1})$ in (13.6.4), (13.6.7) etc.

(c) In the state estimation stage, the division by $C(q^{-1})$ need not be explicitly performed, if computing the control $u(k)$ is the prime objective. The vector $C(q^{-1})\hat{\mathbf{x}}(k|k)$ is obtained directly from (13.6.7); the control $u(k)$ is computed from

$$u(k) = -\mathbf{k}^T(k)C(q^{-1})\hat{\mathbf{x}}(k|k) - (C(q^{-1}-1))u(k).$$

The state estimation and the LQ control scheme are shown in Fig.13.6.1.

Example 13.6.1 Determine $\hat{\mathbf{x}}(k|k)$ for the process:

$$y(k) - 1.7y(k-1) + 0.72y(k-2) = 0.4u(k-4) + 0.8u(k-5) + e(k) - 0.5e(k-1).$$

Here, the size of the state vector (in (13.6.1)) is given by max (degree of $A(q^{-1})$, (degree of $B(q^{-1})$+d), degree of $C(q^{-1})$) = max {2,5,1} = 5. Again,

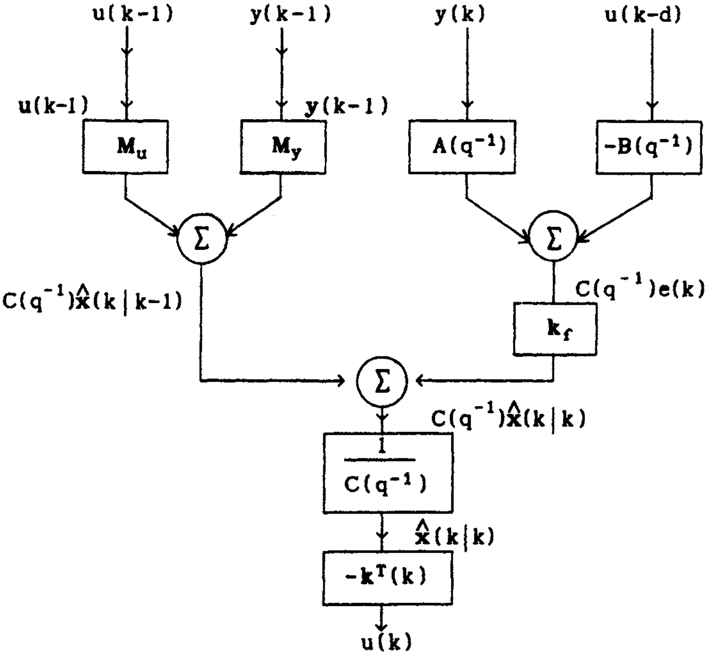


Figure 13.6.1 State estimation for CARMA model and the computation of the LQ control law u(k). $\mathbf{M}_u$ and $\mathbf{M}_y$ are the transmittance matrices, $\mathbf{k}_f$ is the steady state Kalman filter gain, **k**(k) is the control gain.

$$\mathbf{b} = [0\ \ 0\ \ 0\ \ 0.4\ \ 0.8]^T,$$

$$\mathbf{e} = [(-0.5+1.7)\ \ (-0.72)\ \ 0\ \ 0\ \ 0]^T = [1.2\ \ -0.72\ \ 0\ \ 0\ \ 0]^T,$$

and $a_1 = -1.7$ and $a_2 = 0.72$ in **A**. Following (13.6.4),

$$\mathbf{F} = [\mathbf{A}-\mathbf{s}\mathbf{c}]^T = \begin{bmatrix} 0.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Following (13B.13 in Appendix 13B),

$$\mathbf{M}_u = \begin{bmatrix} 0 & 0 & 0 & 0.4 & 0.8 \\ 0 & 0 & 0.4 & 0.6 & -0.4 \\ 0 & 0.4 & 0.6 & -0.2 & 0 \\ 0.4 & 0.6 & -0.2 & 0 & 0 \\ 0.8 & -0.4 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \mathbf{M}_y = \begin{bmatrix} 1.2 & -0.72 & 0 & 0 & 0 \\ -0.72 & 0.36 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The state estimate is now computed using (13.6.7), where

$$k_f = [1 \quad -0.5 \quad 0 \quad 0 \quad 0\,]^T, \text{ and}$$

$$e(k) = y(k) - 1.7y(k-1) + 0.72y(k-2) - 0.4u(k-4) - 0.8u(k-5).$$

u(k) can now be computed as discussed in Example 13.7.1.

13.6.2 State Estimation from CARIMA Model

State estimation for LQ regulator with zero set point

Following (13.5.7 - 13.5.8)

$$\hat{x}(k+1|k) = \tilde{A}\hat{x}(k|k-1) + bu(k) + \tilde{s}e(k), \qquad (13.6.8)$$

$$y(k) = c^T\hat{x}(k|k-1) + e(k). \qquad (13.6.9)$$

The state estimate is given by

$$\hat{x}(k|k) = \hat{x}(k|k-1) + k_f(y(k) - c^T\hat{x}(k|k-1)).$$

Eliminating e(k) in (13.6.8) by substituting from (13.6.9)

$$\hat{x}(k+1|k) = [\tilde{A}-\tilde{s}c^T]\hat{x}(k|k-1) + b\Delta u(k) + \tilde{s}y(k).$$

That is

$$\hat{x}(k|k-1) = [I-q^{-1}F]^{-1}[b\Delta u(k-1) + \tilde{s}y(k-1)], \qquad (13.6.10)$$

where $F = [\tilde{A}-sc^{\sim}]^T$ is the same as in case of the CARMA model (13.6.4), and hence is strictly stable. Introducing the transmittance matrices M_u and M_{yI} for $\Delta u(k-1)$ and $y(k-1)$ respectively,

$$\hat{x}(k|k-1) = \frac{1}{C(q^{-1})}[M_u\Delta u(k-1) + M_{yI}y(k-1)],$$

where

$$\Delta u(k-1) = [\Delta u(k-1) \quad \Delta u(k-2) \ \dots\ ]^T,$$

$$y(k-1) = [y(k-1) \quad y(k-2) \ \dots\ ]^T.$$

Hence the desired state estimate is given by

$$\hat{x}(k|k) = \frac{1}{C(q^{-1})}[M_u\Delta u(k-1) + M_{yI}y(k-1) + k_f(A(q^{-1})y(k)-B(q^{-1})u(k-d))].$$

where

$$k_f = [1 \ c_1 \ c_2 \ \dots\ ]^T.$$

The LQ control law is given by

$$\Delta u(k) = u(k) - u(k-1) = -\mathbf{k}^T(k)\hat{\mathbf{x}}(k|k). \qquad (13.6.11)$$

Remarks

(a) Here, the control law is expressed in terms of Δu because the control *increments* are costed in the cost function (13.5.16).

(b) $\mathbf{F}$, $\mathbf{M}_u$, $\mathbf{y}(k-1)$ and $\mathbf{k}_f$ are the same for the CARMA and the CARIMA models. $\mathbf{M}_{yI}$ is different from $\mathbf{M}_y$ because of $\tilde{\mathbf{s}}$ appearing in (13.6.10) instead of $\mathbf{s}$ in (13.6.3).

State estimation for LQ controller with nonzero set point

The state-space model (13.5.11-13.5.12) for the process (13.5.10)

$$\tilde{A}(q^{-1})\varepsilon(k) = B(q^{-1})\Delta u(k-d) - A(q^{-1})\Delta w(k) + C(q^{-1})e(k),$$

can be expressed in the innovations form:

$$\hat{\mathbf{x}}(k+1|k) = \tilde{\mathbf{A}}\mathbf{x}(k|k-1) + \mathbf{b}\Delta u(k) + \mathbf{w}\Delta w(k+1) + \tilde{\mathbf{s}}e(k),$$

$$\varepsilon(k) \quad = \mathbf{c}^T\hat{\mathbf{x}}(k|k-1) + e(k);$$

the state estimate is given by

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{k}_f(\varepsilon(k) - \mathbf{c}^T\hat{\mathbf{x}}(k|k-1)).$$

Eliminating e(k) from the state-space model,

$$\hat{\mathbf{x}}(k+1|k) = [\tilde{\mathbf{A}} - \tilde{\mathbf{e}}\mathbf{c}^T]\hat{\mathbf{x}}(k|k-1) + \mathbf{b}\Delta u(k) + \mathbf{w}\Delta w(k+1) + \tilde{\mathbf{s}}\varepsilon(k).$$

That is

$$\hat{\mathbf{x}}(k|k-1) = [\mathbf{I}-q^{-1}\mathbf{F}]^{-1}[\mathbf{b}\Delta u(k-1) + \mathbf{w}\Delta w(k) + \tilde{\mathbf{s}}\varepsilon(k-1)],$$

where $\mathbf{F} = [\tilde{\mathbf{A}} - \tilde{\mathbf{e}}\mathbf{c}^T]$ is the same as the $\mathbf{F}$ in the CARMA case (13.6.4), and hence is strictly stable. Introducing the transmittance matrices $\mathbf{M}_u$, $\mathbf{M}_w$, $\mathbf{M}_e$ for $\Delta u(k-1)$, $\Delta w(k)$ and $\underline{\varepsilon}(k-1)$ respectively

$$\hat{\mathbf{x}}(k|k-1) = \frac{1}{C(q^{-1})}[\mathbf{M}_u\Delta u(k-1) + \mathbf{M}_w\Delta w(k) + \mathbf{M}_e\underline{\varepsilon}(k-1)], \qquad (13.6.12)$$

where

$$\Delta u(k-1) = [\Delta u(k-1)\ \Delta u(k-2)\ \ldots]^T,$$

$$\Delta w(k) \quad = [\Delta w(k)\ \Delta w(k-1)\ \ldots\ ]^T, \quad \text{and}$$

$$\underline{\varepsilon}(k-1) \quad = [\varepsilon(k-1)\ \varepsilon(k-2)\ \ldots\ ]^T.$$

Hence the desired state estimate is given by

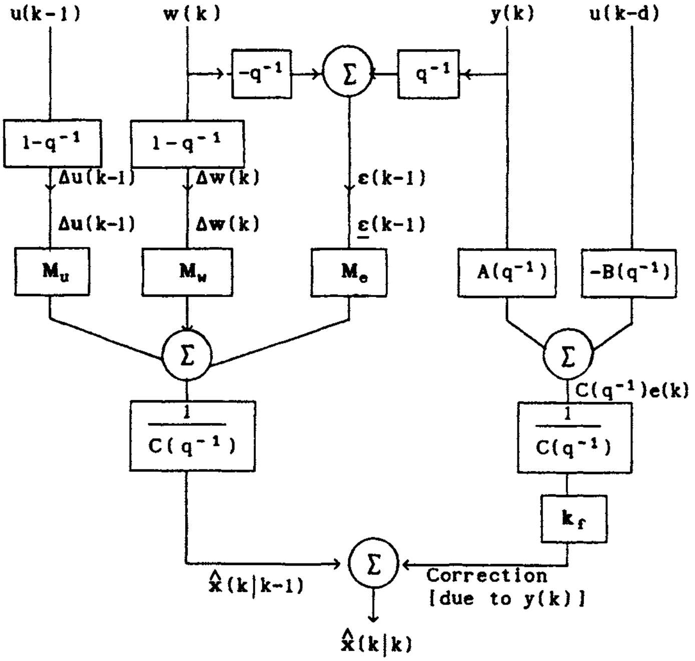


Figure 13.6.2 Estimation of the state $\hat{x}(k|k)$ for a CARIMA model with nonzero set point w(k). For zero set point cases, M_e and $\underline{\varepsilon}(k-1)$ are replaced with M_y and $\underline{y}(k-1)$ respectively. For CARMA models, all $(1-q^{-1})$ blocks disappear. Division by $C(q^{-1})$ need not be explicitly implemented as shown in Fig.13.6.1

$$\hat{x}(k|k) = \frac{1}{C(q^{-1})} [M_u\Delta u(k-1) + M_w\Delta w(k) + M_e\underline{\varepsilon}(k-1)] + K_f e(k)$$

$$= \frac{1}{C(q^{-1})} [M_u\Delta u(k-1) + M_w\Delta w(k) + M_e\underline{\varepsilon}(k-1) + k_f(A(q^{-1})y(k) - B(q^{-1})u(k-d))].$$

Fig.13.6.2 summarizes the state estimation scheme. The LQ control law is given by (13.6.11).

13.7 COMPUTATION OF CONTROL

The stochastic optimal controller comprises a deterministic controller followed by a stochastic state estimator as discussed in Sec.13.4.

Restating (13.2.4 - 13.2.6), for the state-space model (13.6.2 - 13.6.3), the LQ control gain **k**(k) is obtained from the backward recursive solution of the following equations

$$\mathbf{k}^T(k) = (\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(k+1)\mathbf{A}, \qquad (13.7.1)$$

$$\mathbf{P}^*(k) = \mathbf{P}(k+1) - \mathbf{P}(k+1)\mathbf{b}(\lambda+\mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(k+1), \qquad (13.7.2)$$

$$\mathbf{P}(k) = \mathbf{Q} + \mathbf{A}^T\mathbf{P}^*(k)\mathbf{A}, \qquad (13.7.3)$$

where $\mathbf{Q} = \mathbf{c}\mathbf{c}^T$ and $\mathbf{P}(k+N) = \mathbf{Q}$. In case of models derived from CARIMA representation, (i.e. for the model (13.5.7 - 13.5.8)), $\tilde{\mathbf{A}}$ replaces **A** in (13.7.1) and (13.7.3); all other terms remain the same.

At every sampling time k, the control gain **k**(k) is computed using the covariance matrix **P**(k+1). The computation of **P**(k+1) involves two main considerations: (i) Control horizon, (ii) Implementation using the principle of duality.

Remark: The computation of **k** does not depend on whether the set point is zero or not, as structurally the cost function minimized remains the same (as discussed in Sec.13.5.2).

13.7.1 Control Horizons

The control gain **k**(k) depends on the control horizon considered. The different classes of horizons (shown in Fig.13.7.1) are as follows.

(a) *Infinite horizon*: In this case the terminal covariance considered is $\mathbf{P}(\infty) = \mathbf{Q}$. At each sampling instant **P**(k+1) is calculated through backward recursion of (13.7.2) and (13.7.3) to convergence, starting from the terminal stage $\mathbf{P}(\infty)$.

(b) *Adaptive infinite horizon*: Here the recursion (13.7.2 - 13.7.3) is carried out a finite number of times (usually once) starting with the covariance obtained at the last sampling instant.

(c) *Receding finite horizon*: The terminal covariance considered is $\mathbf{P}(k+N) = \mathbf{Q}$, and N number of backward recursions of (13.7.2 - 13.7.3) are executed to compute **P**(k+1). If the process parameters remain unchanged, the resulting control law will be time invariant.

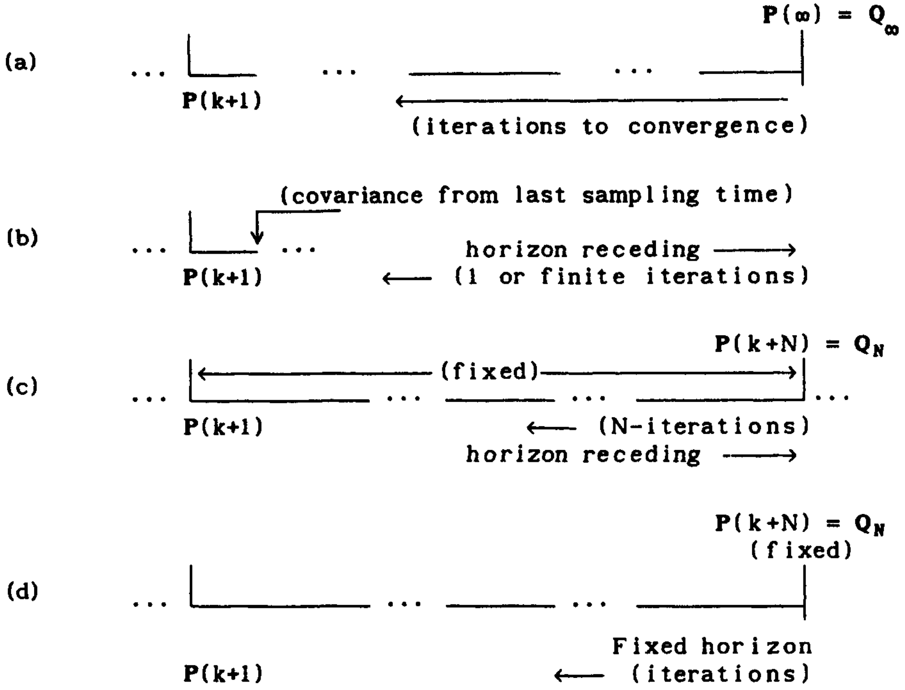


Figure 13.7.1 Different horizons for backward iterations of covariance matrix for the LQ controller:
(a) Infinite horizon, (b) Adaptive infinite horizon,
(c) Receding finite horizon, (d) Fixed horizon.

(d) *Fixed horizon*: P(k+N) = Q is considered as the terminal covariance and k is assumed to lie within fixed time instants k and (k+N). The backward recursion (13.7.2 - 13.7.3) are performed starting from P(k+N) = Q to P(k+1), and the Kalman gain **k**(i) is calculated at each recursion and stored. As time progresses from k to (k+N) the corresponding precomputed **k**(i) are used to calculate the control law.

13.7.2 Implementation based on the Principle of Duality

The principle of duality between the optimal filter and the optimal controller is discussed in Sec.13.4. The two stages of the Riccati equation (13.7.2) and (13.7.3) are duals of the *measurement update* and the *time update* respectively of the Kalman filter. Thus computation of the gain **k**(k) will require two subroutine calls: (i) the measurement update subroutine (which is also used for the least squares parameter estimation), and (ii) the time update subroutine. For numerical stability and computational efficiency, U-D factorization of the covariance matrix is used. The measurement update is discussed in Appendix 3A.

Bierman (1977, p.124) discusses a general procedure for the U-D time-update (13.7.3) based on matrix operations. However, a much simpler vector implementation is possible in the present case by the reformulation of the decomposition problem as follows. Given the factors, $\mathbf{U}^*$ and $\mathbf{D}^*$, the problem is to compute the updated factors of **P** in

$$\mathbf{P}(k) = \mathbf{Q} + \mathbf{A}^T\mathbf{P}^*(k)\mathbf{A}, \qquad \mathbf{P}^*=\mathbf{U}^*\mathbf{D}^*\mathbf{U}^{*T},$$

where

$$\mathbf{Q} = \mathbf{c}\mathbf{c}^T = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & 0 & \\ 0 & & & \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [1 \; 0 \; \dots \; 0],$$

$$= \mathbf{g}q\mathbf{g}^T \text{ (say);}$$

that is

$$\mathbf{g} = [1 \quad 0 \quad \dots \; 0]^T, \text{ and } \quad q = 1.$$

So

$$\mathbf{P} = \mathbf{g}q\mathbf{g}^T + \mathbf{A}^T\mathbf{P}^*\mathbf{A} = [\mathbf{g} \quad \mathbf{A}^T\mathbf{U}^*] \operatorname{diag}[q \quad \mathbf{D}^*] \; [\mathbf{g} \; \mathbf{A}^T\mathbf{U}^*]^T,$$

$$= \mathbf{W}\bar{\mathbf{D}}\mathbf{W}^T \text{(say)} \qquad (13.7.4)$$

$$= \mathbf{U}\mathbf{D}\mathbf{U}^T.$$

The updated **U** and **D** factors of **P** can be computed from $\mathbf{W}\bar{\mathbf{D}}\mathbf{W}^T$ using the modified weighted Gram-Schmidt algorithm. In the present context, since **A** is in observable canonical form, **W** has a strictly zero lower triangular part as shown below.

Example: Let

$$\mathbf{A}^T = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \text{ and } \; \mathbf{U}^* = \begin{bmatrix} 1 & u_{12}^* & u_{13}^* \\ & 1 & u_{23}^* \\ & & 1 \end{bmatrix}. \qquad (13.7.5)$$

Hence

$$\mathbf{W} = \mathbf{A}^T\mathbf{U}^* = \begin{bmatrix} 1 & -a_1 & (-a_1u_{12}^*-a_2) & (-a_1u_{13}^*-a_2u_{23}^*-a_3) \\ 0 & 1 & u_{12}^* & u_{13}^* \\ 0 & 0 & 1 & u_{23}^* \end{bmatrix}. \qquad (13.7.6)$$

Since **W** has strictly zero lower part, construction of **W** and subsequent orthogonalization becomes easier with reduced computational load.

U-D time-update, its vector formulation, and FORTRAN implementation is given in Appendix 13C.

Remarks: Equivalence between LQ-Control and GPC
The deterministic LQ-control with finite horizon (given by for example (13.5.14) and (13.5.16)), and the generalized predictive control (GPC) discussed in Sec.12.6 minimize the same cost function and produce identical control actions, although the LQ-controller is state-space model based and is solved through backward recursions whereas GPC is input-output model based and is solved through predictions in the forward direction. Consequently the stability properties for both the control methods are the same. Numerically stable implementations are available for both the algorithms. For further discussions see Clarke *et al* (1987).

Example 13.7.1 Compute the control law minimizing J_1, given by (13.5.13a) with λ = 0.5 and N = 12, for the process considered in Example 13.6.1.

Two subroutines will be used to calculate **k**(k):
(i) U-D measurement update (Appendix 3A) to solve (13.7.2),
(ii) U-D time update (Appendix 13C) to solve (13.7.3).

Step 1: Initialize with **P**(k+12) = **cc**^{T}, where **c** = $[1\ 0\ 0\ 0\ 0]^T$. That is, initialize with the U-D factors,

$$D(k+12) = [1\ 0\ 0\ 0\ 0]^T;$$

all elements of the vector U(k+12) are initialized as zeros.
Step 2: Starting with U(k+12) and D(k+12), perform U-D measurement update to produce U*(k+11) and D*(k+11).
Step 3: Perform U-D time update to produce updated U(k+11) and D(k+11) vectors.
Step 4: Iterate through (steps 2 to 3) until U*(k) and D*(k)

are produced.

Remarks: The computation of the gain **k**(k) in (13.7.1) is not performed within the iterative loop. **k**(k) emerges from the last U-D measurement update as $\mathbf{k}_u(k) = \mathbf{P}(k+1)\mathbf{b}$, which is the *unweighted* control gain. The measurement update routine also produces the scalar $(\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})$ as ALPHAJ (See Appendix 3A). So the computation of **k**(k) proceeds as follows.

Step 5: Compute k(k) as

$$\mathbf{k}(k) = \frac{\mathbf{k}_u^T(k)\mathbf{A}}{\lambda + \mathbf{b}^T\mathbf{P}(k+1)\mathbf{b}}, \qquad \mathbf{k}_u(k) = \mathbf{P}(k+1)\mathbf{b},$$

In the present case,

$$\mathbf{k}(k) = [1.6757 \quad 1.6243 \quad 1.5077 \quad 1.3039 \quad 0.9846]^T.$$

Step 6: The control $u(k) = -\mathbf{k}^T(k)\hat{\mathbf{x}}(k|k)$, is computed from

$$C(q^{-1})u(k) = -\ \mathbf{k}^T(k)[\mathbf{M}_u u(k-1) + \mathbf{M}_y y(k-1) + \mathbf{k}_f e(k)],$$

where all the terms excepting k(k) are obtained as in Example 13.6.1. Hence the control law works out as

$$\begin{aligned} u(k) \; &= -0.8636y(k) + 0.2368y(k-1) - 0.8092u(k-1) \\ &\quad - 0.9916u(k-2) - 1.2935u(k-3) - 0.998\ u(k-4); \end{aligned}$$

Remark: If y(k) and hence e(k) is ignored,

$$\begin{aligned} u(k) \; &= -\ 0.8413y(k-1) + 0.6218y(k-2) \\ &\quad -\ 0.8092u(k-1) - 0.9916u(k-2) - 1.2935u(k-3) \\ &\quad - 1.3433u(k-4) - 0.6909u(k-5). \end{aligned}$$

13.7.3 Implementation Aspects and Features

Some of the main features ascribing numerical stability, computational efficiency and robustness to the present algorithm are discussed here.

Riccati equation

The solution of the Riccati equations (13.7.2 - 13.7.3) using U-D factorization (discussed in Sec.13.7.2) is found to be particularly robust against round off errors and other instabilities. Although $UD^{1/2}$ corresponds directly to the square root of the covariance matrix (since $\mathbf{P} = \mathbf{UDU}^T$) and hence having all the advantages of square-root propagation, the square roots are never explicitly evaluated.

The U-D factorization for the covariance time-update in the present case is particularly short and efficient due to vector implementation.

Control cost factor (λ)

If λ, the scalar cost factor on control in (13.5.13-13.5.16), is zero and if the leading element of **b** vector is also zero, numerical difficulties in the U-D measurement update of the Riccati equation result, because the Agee-Turner factorization used (Bierman, 1977, p.78) fails under such conditions. However, this problem can be avoided by assigning a vanishingly small value to λ (for example, a number near the floating point zero of the computer). Note that with vanishingly small λ, the LQ controller can handle nonminimum-phase processes, unlike the generalized minimum variance controller.also minimizing the cost (12.2.6).

The value of λ can have a wide range. Larger λ results in slower (or more sluggish) control response.

Implementational simplicity

(i) The implementation does not require explicit construction of the matrix **A** (or $\tilde{\mathbf{A}}$ for the CARIMA model), the parameters a_i (or $\tilde{a}_i$) can be used directly. Similarly $\mathbf{P}^*$ and **P** matrices also need not be formed explicitly; only their factorized U-D components expressed in vectors are used.
(ii) If the solution of the Riccati equation requires more than one recursion, only the covariances are updated, i.e. cycling through equations (13.7.2) and (13.7.3) inside the recursive loop without calculating the Kalman control gain. On exit, the gain, which is available as a by-product of the covariance update, appears as $\mathbf{P}(k+1)\mathbf{b}$ from which the final control gain (13.5.36) is calculated, where $(\lambda+\mathbf{b}^T\mathbf{P}(k+1)\mathbf{b})^{-1}$ is also available from the covariance update.

Mismodelled process dynamics

Due to over-parameterization (which may even be due to fast sampling), there may be unstable common (or nearly common) factors between the estimated $A(q^{-1})$ and $B(q^{-1})$ polynomials. These mismodelled dynamics may show up in unboundedness of the diagonal elements of the control covariances. This can be arrested by using hard constrains on the D-factors and the algorithm can still be operational.

Computation time

Unlike algorithms involving solution of the Diophantine equations or spectral factorization, the solution of the Riccati equation requires a fixed time for adaptive infinite and fixed horizon cases.

The execution time for the present LQ control law is approximately equivalent to 5 U-D filter updates: 1 parameter estimation, 2 U-D Riccati measurement updates, 1 U-D time update and 2 state estimation (for non-zero set point) using transmittance matrices. The computation time can be further reduced by using lattice filter mechanizations.

13.7.4 Self-tuning control

So far the discussions have been confined to the design of the LQ controller, when the process parameters are known. In practice, the process parameters are usually unknown. In such cases, the principle of self-tuning can be used; that is, the parameters of the process model are estimated from the input and output data, and assuming the estimated parameter values to be true, the state is estimated and the LQ control gain is computed. In other words, the principle of certainty equivalence (introduced in Sec.12.2.1) is invoked in computing the control action. The term self-tuning implies that asymptotically as the estimated parameters reach true values, the computed control law will be the same as the optimal control, which could be produced if the actual parameters were known.

13.8 SIMULATION STUDIES

LQ control of single-input single-output processes is discussed. A CARIMA process model is considered. The parameters are estimated using the recursive least squares estimator, with no forgetting. No prior knowledge of the parameters is assumed. The cost function minimized is J_3' given by (13.5.16); an adaptive infinite control horizon (Fig.13.7.1) is used for computation of the control input u,

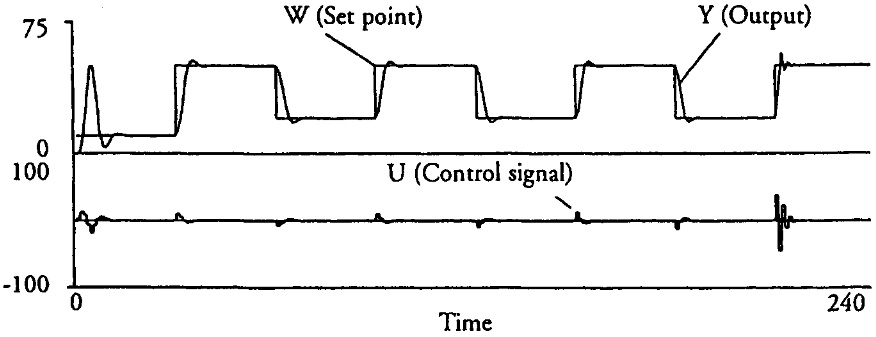
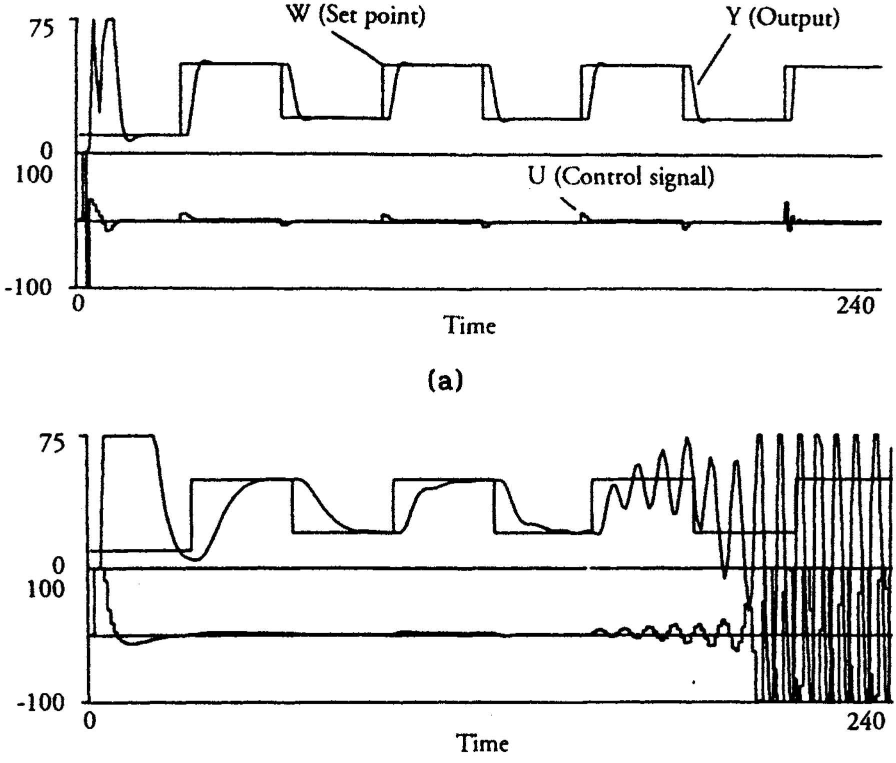


Figure 13.8.1 LQ control of a nonminimum-phase process.

which is constrained to lie within +100 and −100 values. The cost factor λ is varied from 2 to 0 (where 0 is assumed to be vanishingly small). Additive step changes ξ_y and ξ_u to output y and input u respectively are applied to simulate external disturbances acting on the process. Simulation runs over 240 samples are shown in Figs.13.8.1 to 13.8.3, and are summarized in Table 13.8.1. These results are extracted from Clarke, Kanjilal, and Mohtadi (1985b).

Exercise 1: A nonminimum-phase process is controlled. For $\lambda = 0$, the unstable zero in $B(q^{-1})$ at −2 is reflected inside the unit circle, resulting in a closed-loop pole at −1/2; although the control signal is active, the stability of the loop is maintained as shown in Fig.13.8.1.

Exercise 2: An open-loop stable minimum-phase process is considered; the performances of the LQ controller and the generalized minimum variance controller (GMV) (12.2.6) is studied, when the time-delay is underestimated as 1 instead of the actual 3. Due to underestimation of the time-delay and the leading coefficients of $B(q^{-1})$ being zero (or estimated to be too low), the estimated process is rendered nonminimum-phase. With high cost (i.e. high λ) on control GMV can still control the process but with $\lambda = 0$, the GMV which becomes the minimum variance control, fails. The LQ control produces a stable control even when $\lambda = 0$. The results are shown in Fig.13.8.2.



(a)

(b)

Figure 13.8.2 Comparative performance of controllers for over-parameterized process model and underestimated time-delay (a) LQ control, (b) GMV control (Sec.12.2.2).

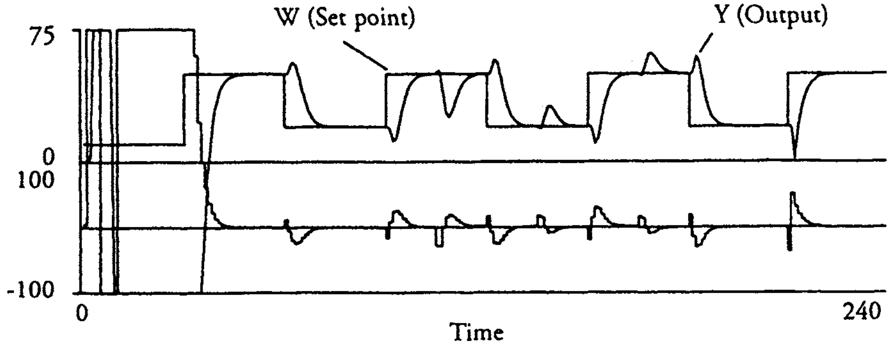


Figure 13.8.3 LQ control of an over-parameterized, unstable, nonminimum-phase process with over-parameterized model and underestimated time-delay.

Table 13.8.1 Summary of simulation exercises

Exercise	Simulated Process	Remarks
Exercise 1 (Fig. 13.8.1)	$B(q^{-1})=q^{-1}(0.4+0.8q^{-1})$ $A(q^{-1})=1-1.7q^{-1}+0.72q^{-1}$	Nonminimum-phase process, correct parameterization, correct time-delay. λ varied between 2 to 0 stable control.
Exercise 2 (Fig. 13.8.2)	$B(q^{-1})=q^{-3}(1+0.5q^{-1})$ $A(q^{-1})= 1 - 0.9q^{-1}$	Minimum-phase process, correct parameterization, underestimated time-delay (1 instead of 3). GMV stable for high λ, but fails for $\lambda = 0$ (as becomes MV control). LQ control stable for $\lambda = 2$ to 0.
Exercise 3 (Fig. 13.8.3)	$B(q^{-1})=q^{-2}(0.5-0.8q^{-1})$ $A(q^{-1})=1 - 3q^{-1} + 2q^{-2}$	Open loop unstable nonminimum-phase process. Over estimated model order underestimated time delay (1 instead of 2). $\lambda = 2$ to 0. LQ control stable.

Exercise 3: Here, control of a nonminimum-phase, open-loop unstable process is considered, when the controller is based on an over-parameterized model with underestimated time-delay (i.e 1 instead of the actual time-delay of 2). The control disturbance ξ_u of magnitude +0.5, −0.25 and −0.25 is applied at 45, 140 and 165 sampling instants respectively. The estimator has three $\hat{a}_i$ and four $\hat{b}_i$ parameters, although only two $\hat{a}_i$ and three $\hat{b}_i$ parameters are required for a correctly parameterized model with underestimated time-delay. The plant has a negative going nonminimum-phase characteristic which is responsible for the initial movement of the plant in the negative direction at positive step changes in the set point. However, the LQ controller provides good response both in set-point following as well as in disturbance rejection.

13.9 CONCLUSIONS

The state-space formulation of LQ control has been studied; a linear quadratic cost criterion is considered, which is minimized over a predictive horizon. Control of both time invariant and time varying processes have been discussed.

The stochastic LQ control has been shown to comprise two separate components, namely a linear estimation problem and a quadratic control problem for a linear deterministic process; the former can be solved using the Kalman filter to produce optimal state-estimates, whereas the deterministic LQ controller works out to be dual of the optimal estimator.

A method of state estimation has been presented, which is based on the state-space innovations model of the input-ouput representation and a steady-state Kalman filter. Numerically stable and computationally efficient implementations have been discussed.

Irrespective of the duality between the optimal control and the optimal estimator, structurally an optimal control problem is better posed. This is because optimal control is a closed-loop problem and there is a specific set-point trajectory to be followed, whereas an optimal estimator, which is an open-loop problem, has no such guideline. As a result, an optimal estimator (or filter), by itself, can be comparatively sensitive to improper assumptions (e.g. of noise covariances etc.) in the implementation, whereas when used within the optimal controller, the sensitivity of the controller to the same assumptions will be much less.

Optimal predictors, discussed in earlier chapters, also have the same configurational disadvantage. Again, following the same arguments, the long range predictive control methods (discussed in Chapter 12) are significantly robust due to the presence of the inherent feedback loop, and are relatively insensitive to the quality of the multistep predictions.

The equivalence between the performances of the deterministic state-space LQ control and the generalized predictive control shows that for identical processes when the same cost functions are optimized, the same control action is produced irrespective of the design approaches.

REFERENCES

Remarks: Linear quadratic control is a widely covered subject. The texts [1,5,11,14,15,16] provide a detailed study of the theoretical as well as implementation aspects. There are many papers on this subject: [2,9]. Optimal estimation of states is studied in the seminal paper [10], where the principle of duality between the estimator and the controller is also introduced. Design of the optimal controller through *Dynamic programming* is treated in [3]. The LQ controller designed in a stochastic environment from input-output models appears in [6,7,12,13], where the computational aspects are also discussed. Detailed treatment of computational aspects appears in [4]. Comparative study of LQ control and the generalized predictive control methods features in [8].

[1] Anderson, B.D.O., and J. Moore (1990): *Optimal Control: Linear Quadratic Methods*, Prentice Hall, Englewood Cliffs, New Jersey.

[2] Athans, M. (1971): 'The role and use of the stochastic Linear-Quadratic-Gaussian problem in control system design', *IEEE Trans. on Auto. Control*, AC-16 (6), 529-552.

[3] Bellman, R. (1957): *Dynamic Programming*,. Princeton University Press, Princeton, N.Y.

[4] Bierman, G. J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[5] Bryson, A.E., and Y.C. Ho (1969): *Applied Optimal Control*, Hallstead, New York.

[6] Clarke, D.W., P.P. Kanjilal, and C. Mohtadi (1985a): 'A Generalized LQG Approach to Self-tuning Control, Part I: Aspects of Design', *Int. J. Control*, 41, 1509-1523.

[7] Clarke, D.W., P.P. Kanjilal, and C. Mohtadi (1985b): 'A Generalized LQG Approach to Self-tuning Control, Part II. Implementation and Simulation', *Int. J. Control*, 41, 1525-1544.

[8] Clarke, D.W., C. Mohtadi, and P.S. Tuffs (1987): 'Generalized predictive control - Part II, extensions and interpretations', *Automatica*, 23(2), 149-160.

[9] IEEE Trans. on Automatic Control (1971): Special issue on 'The Linear Quadratic Gaussion Estimation and Control Problem'.

[10] Kalman, R.E. (1960): 'A new approach to linear filtering and prediction problems', *Trans. ASME, Journal of Basic Engineering*, 82 D, 35-45.

[11] Kwakernaak, H., and R. Sivan (1972): *Linear Optimal Control Systems*, Wiley, New York.

[12] Lam, K.P. (1982a): 'Design of stochastic discrete time linear optimal regulators, Part I: Relationship between control laws based on a time series approach', *Int. J. Systems Sci.*, 13, 979-1000.

[13] Lam, K.P. (1982b): 'Design of stochastic discrete time linear optimal regulators, Part II: Extension and computational procedures', *Int. J. Systems Sci.*, 13,

1001 - 1011.

[14] Maybeck, P.S. (1982): *Stochastic Models, Estimation and Control*, Vol.3, Academic Press, New York.

[15] Meditch, J. S. (1979): *Stochastic Optimal Linear Estimation and Control*, McGraw-Hill, New York.

[16] Strejc, V. (1981): *State Space Theory of Discrete Linear Control*, Wiley, Chichester.

CHAPTER 14

SMOOTHING AND FILTERING

Data = Signal + noise
(information) (contamination)

Some data are born noisy, some data pick up noise in transportation, while some data have noise thrust upon them due to improper processing! The information in the data has to be separated from the contaminations before use.

14.1 INTRODUCTION

The characteristic features of the information contained in the data are (a) the frequency components present, (b) the pattern in the data, (c) the real-timeliness of the data etc. The data represent the behaviour of the underlying process in terms of such features. In practice, the information in data is rarely devoid of noise contaminations. The noise may be inherently associated with the data, or it may be linked with the data at a subsequent stage. For example, (i) any data obtained from empirical measurements (say, the sinter strength measurement discussed in Sec. 5.6.2) are generated noisy because of the empirical nature of the measurement; (ii) the measurement of the fetal ECG through maternal ECG, is a case where noise is picked up in transportation; again (iii) the phase-shift or time-lag introduced in a data sequence due to exponential smoothing or low-pass filtering (see Fig.4.2.2) is a case of noise being thrust upon the data.

The noise associated with the data should be either eliminated or its influence should be substantially reduced when the data are to be used for identification, prediction or control purposes. Both smoothing and filtering are aimed at minimizing the effects of noise in the data. It is important that the characteristic features of the information contained in the data are not affected in the process of smoothing and filtering.

By definition, *smoothing* implies estimation of past values based on all the information available up to the present time. Scientists belonging to different disciplines often use different methods for smoothing (see for example Sec.4.2.1). In this chapter two classes of approaches are presented, namely optimal smoothing and bidirectional (low-pass) filtering. Optimal smoothing algorithms designed in state-space framework, are discussed in Sec.14.2. Three useful classes of optimal smoothers characterized by the smoothing interval, the point at which the smoothed value is desired and the lag at which smoothing is to be performed, are studied.

Bidirectional smoothing, treated in Sec.14.3, offers a comparatively simpler smoothing approach; it is applicable when the frequencies contained in the signal are lower than those contained in the noise.

Orthogonal transformation, offers an alternative approach to smoothing and filtering, where separation of the signal is achieved through separation of the orthogonal components; so it is conceptually different from frequency based smoothing or filtering. This subject is introduced in Sec.14.4. The applications of SVD in smoothing, pattern estimation and selective filtering are explored in Sec.14.5. A case study on the extraction of fetal ECG from maternal ECG signal is presented in Sec.14.6.

14.2 OPTIMAL STATE-SPACE SMOOTHING

Smoothing algorithms based on state-space models of the process are studied in this section. The state-space framework permits the use of many well-studied and established algorithms (like the Kalman filter etc.), for which numerically stable and computationally efficient implementations are available. A detailed study of optimal smoothing is beyond the scope of this book; the prime concepts and some of the algorithms are discussed here.

Optimal smoothing concerns estimation of past values of the variables of interest, based on the available information:

$$\hat{x}(k|k+i) = E\{x(k)|y(0),y(1),\ldots,y(k+i)\},$$

that is, given the noisy measurements y(0),...,y(k+i), the objective is to determine the optimally smoothed estimate for x(k) in state smoothing or for $\hat{y}(k)$, in data smoothing. In comparison with the estimation $\hat{x}(k|k)$, it is expected that $\hat{x}(k|k+i)$, produced with the incorporation of additional i-measurements, will be more representative though at the cost of additional computation and complexity.

Smoothing problems belong to three main categories (see Fig.14.2.1) as follows.

(1) *Fixed-interval smoothing* implies estimation of $\hat{x}(k|N)$, for $0 \le k \le N$, i.e. k lying within the finite time interval, 0 to N, where N is the fixed final time.

(2) *Fixed-point smoothing* implies estimation of $\hat{x}(k|k+i)$, where k is a fixed point in time and k+i stands for all subsequent points in time.

(3) *Fixed-lag smoothing* implies estimation of $\hat{x}(k|k+L)$ where k is any point in time and k+L is a point L (constant) steps ahead of k.

The optimally smoothed estimate is also the minimum variance estimate, the cost minimized being, for example in case of fixed interval smoothing:

$$\begin{aligned} J &= \min_{\hat{x}(k|N)} \; E\{[x(k)-\hat{x}(k|N)]^T[x(k)-\hat{x}(k|N)] \;|y(k):k = 0,1,\ldots,N\} \\ &= \min_{\hat{x}(k|N)} \mathrm{Tr}\, E\{[x(k)-\hat{x}(k|N)]\;[x(k)-\hat{x}(k|N)]^T|y(k):k = 0,1,\ldots,N\}. \end{aligned} \tag{14.2.1}$$

Different recursive algorithms have been proposed for solving the optimal smoothing problems (Anderson and Moore, 1979), some of which are presented here.

Problem formulation

Consider the linear discrete-time process model

$$x(k+1) = A(k+1|k)x(k) + s(k+1|k)w(k), \tag{14.2.2a}$$

$$y(k) \;= c^T(k)x(k) + v(k), \tag{14.2.2b}$$

where x is nx1 state vector, y is a scalar measurement; A is an nxn matrix, and c and s are nx1 vectors. {w(k)} and {v(k)} are independent, zero-mean, white noise sequences:

$$E\{w(k)w(\tau)\} = \begin{cases} Q(k), & k=\tau, \\ 0, & k\neq\tau, \end{cases} \qquad E\{v(k)v(\tau)\} = \begin{cases} R(k), & k=\tau, \\ 0, & k\neq\tau. \end{cases}$$

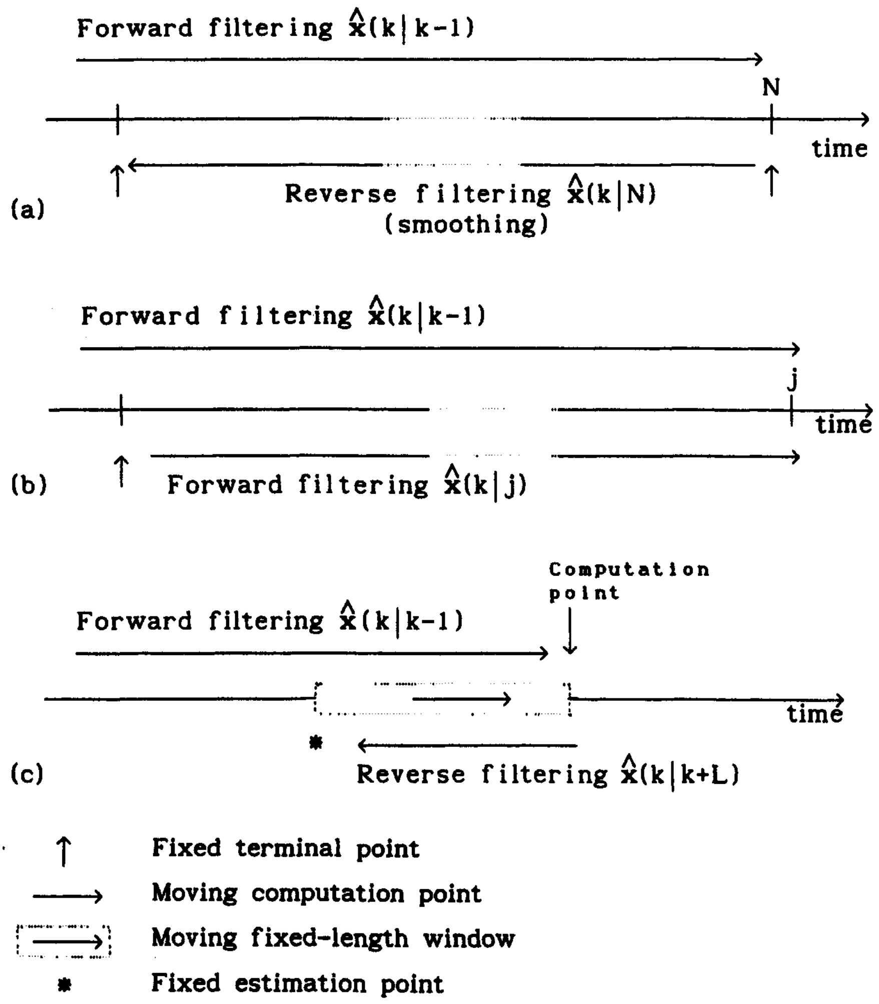

Forward filtering $\hat{x}(k|k-1)$
N
time
(a)
Reverse filtering $\hat{x}(k|N)$
(smoothing)
Forward filtering $\hat{x}(k|k-1)$
j
time
(b)
Forward filtering $\hat{x}(k|j)$
Computation point
Forward filtering $\hat{x}(k|k-1)$
time
(c)
Reverse filtering $\hat{x}(k|k+L)$
Fixed terminal point
Moving computation point
Moving fixed-length window
Fixed estimation point

Figure 14.2.1 Optimal smoothing schematic.
(a) Fixed-interval smoothing: Both the estimation and the computation time-points are fixed at N. Smoothed estimates for all points within the interval are produced.
(b) Fixed-point smoothing: The estimation time is fixed whereas the computation time-point moves forward with time.
(c) Fixed-lag smoothing: Both the computation and the estimation points move forward with fixed mutual distance, the estimation produced after minimum delay of L samples.

It will be seen that the optimal filter (Kalman filter, Sec.6.6) is an integral part of the optimal smoothing algorithms. For the process (14.2.1 - 14.2.2) the Kalman filter equations can be restated as follows.

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{k}(k)(y(k) - \mathbf{c}^T\hat{\mathbf{x}}(k|k-1)), \tag{14.2.3a}$$

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}\hat{\mathbf{x}}(k|k), \tag{14.2.3b}$$

$$\mathbf{k}(k) = \mathbf{P}(k|k-1)\mathbf{c}(\mathbf{c}^T\mathbf{P}(k|k-1)\mathbf{c} + R(k))^{-1}, \tag{14.2.4}$$

$$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{k}(k)\mathbf{c}^T]\mathbf{P}(k|k-1), \tag{14.2.5}$$

$$\mathbf{P}(k+1|k) = \mathbf{A}\mathbf{P}(k|k)\mathbf{A}^T + \mathbf{s}Q(k)\mathbf{s}^T, \tag{14.2.6}$$

with the initial conditions $\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0$ and $\mathbf{P}(0|-1) = \mathbf{P}_0$; the arguments of $\mathbf{A}$, $\mathbf{c}$ and $\mathbf{s}$ are same as in (14.2.2a - 14.2.2b).

The objective is to compute smoothed estimate of the state $\mathbf{x}(k)$: $\hat{\mathbf{x}}(k|k+i)$ for $i \geq 1$. The smoothed estimate of the measurement $y(k)$ is obtained as $\hat{y}(k|k+i) = \mathbf{c}^T(k)\hat{\mathbf{x}}(k|k+i)$.

14.2.1 Fixed-interval Smoothing

Fixed-interval smoothing involves bidirectional or forward and backward filtering. There are two main approaches.

(1) The optimal (Kalman) filter is run in the forward direction in time over the data set for the whole interval and the output of the forward filter is used by a backward filter which is run from the end terminal to the point of interest to obtain the smoothed estimate. In fact the backward filter provides correction to the Kalman estimates to produce smoothed estimates.

(2) Two separate optimal filters are operated in the forward and in the backward directions in time. The desired smoothed estimate is obtained as a weighted sum of the estimates produced by the forward and the backward filters (Fraser and Potter, 1969).

The present studies are based on the former approach.

Rauch-Tung-Streibel (RTS) algorithm

One of the classic approaches to fixed-interval smoothing was due to Rauch, Tung and Streibel (1965):

$$\hat{x}(k|N) = \hat{x}(k|k) + L(k)[\hat{x}(k+1|N) - \hat{x}(k+1|k)], \tag{14.2.7}$$

$$P(k|N) = P(k|k) + L(k)[P(k+1|N) - P(k+1|k)]L^T(k), \tag{14.2.8}$$

$$L(k) = P(k|k)A^T(k+1|k)P^{-1}(k+1|k), \tag{14.2.9}$$

for k = N-1, N-2,..., 0,

where $P(k+1|k)$ is the covariance of the prediction error $\tilde{x}(k+1|k) = x(k+1)-\hat{x}(k+1|k)$, and $P(k|N)$ is the covariance of the fixed-interval smoothing error $\tilde{x}(k|N) = x(k)-\hat{x}(k|N)$.

The main disadvantages of the RTS algorithm are computational and numerical problems connected with (i) the inversion of $P(k+1|k)$ in (14.2.9) which may be ill-conditioned, and (ii) the differencing of two positive (semi)definite matrices in (14.2.8).

Modified Bryson-Frazier (MBF) algorithm

The modified Bryson-Frazier algorithm is a numerically stable and computationally efficient formulation for the fixed-interval smoothing (Bryson and Frazier, 1963, Bierman, 1977, p. 223). The basic idea is (a) to express the smoothed estimate of the state and the covariance in terms of two adjoint or intermediate variables (say, λ and Λ), and (b) to perform the recursive backward filtering using the adjoint variables instead of the states and the covariances. The mBF algorithm is stated as follows

$$\hat{x}(k|N) = \hat{x}(k|k-1) + P(k|k-1)\lambda(k|N), \tag{14.2.10}$$

$$P(k|N) = P(k|k-1) - P(k|k-1)\Lambda(k|N)P(k|k-1), \tag{14.2.11}$$

where k = N-1, N-2,..., 1, the nx1 adjoint vector λ and the nxn adjoint matrix Λ satisfy the following recursive equations:

$$\lambda(k|N) = A_a{}^T\lambda(k+1|N) + cf\hat{y}(k|k-1), \tag{14.2.12}$$

$$\Lambda(k|N) = A_a{}^T\lambda(k+1|N)A_a + cfc^T, \tag{14.2.13}$$

where nxn matrix A_a and the scalar f are defined with appropriate arguments as

$$A_a(k+1|k) = A(k+1|k)[I-k(k)c^T(k)], \tag{14.2.14}$$

$$f(k) = (c^T(k)P(k|k-1)c(k) + R(k))^{-1}, \tag{14.2.15}$$

$$\tilde{y}(k|k+1) = y(k) - c^T(k)\hat{x}(k|k+1),$$

$k = N, N-1, .., 1$, and the terminal conditions are $\lambda(N+1|N) = 0$ and $\Lambda(N+1|N) = 0$. $\mathbf{k}(k)$ and $\mathbf{P}(k|k-1)$ are the Kalman filter gain and the covariance matrix respectively corresponding to the forward filtering (14.2.3 - 14.2.6). $\Lambda(k|N)$ is the covariance of $\lambda(k|N)$:

$$\Lambda(k|N) = E\{\lambda(k|N)\lambda^T(k|N)\},$$

$\lambda(k|N)$ is zero mean.

Different designs of smoothing algorithms based on adjoint or intermediate variables have been proposed (see for example Watanabe (1986), the main objectives being improved numerical stability and computational efficiency.

Special features

(a) Duality with Kalman filter

The backward recursion of the adjoint variables is dual of the Kalman filter computations; Equations (14.2.12) and (14.2.13) are dual of (14.2.3) and (14.2.6) respectively.

(b) Implementation aspects

The mBF algorithm involves the following steps:

(1) Perform forward filtering (14.2.3 - 14.2.6) over the specified interval, $k = 0, 1, \ldots, N$.

(2) Perform backward recursions (14.2.12 - 14.2.13) to compute the adjoint variables $\lambda(k|N)$ and $\Lambda(k|N)$.

(3) Compute the smoothed estimate $\hat{\mathbf{x}}(k|N)$ using (14.2.10). Computation of the error covariance $\mathbf{P}(k|N)$ in (14.2.11) is optional.

The duality with Kalman filtering (or often more directly with optimal control), permits use of stable and efficient implementations, as illustrated in the following example.

Example: Consider an AR process:

$$y(k) + a_1 y(k-1) + \ldots + a_n y(k-n) = v(k),$$

where $\{v(k)\}$ is a sequence of uncorrelated equation error. The objective is to compute fixed interval smoothed measurement sequence $\{y(k)\}$, $k = 0, 1, \ldots, N$.

Let the corresponding state-space model be given by

$$\mathbf{x}(k+1) = \mathbf{A}(k+1|k)\mathbf{x}(k) + \mathbf{s}(k+1|k)\mathbf{w}(k),$$

$$y(k) \;\; = \mathbf{c}^T(k)\mathbf{x}(k) + v(k),$$

where

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 & \ldots & 0 \\ -a_2 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n & 0 & 0 & \ldots & 0 \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ 0 \\ -a_n \end{bmatrix},$$

$$\mathbf{c} = [1 \quad 0 \ldots 0]^T, \text{ and } \quad v = w.$$

If $\quad \mathbf{k} = [k_1 \; k_2 \ldots k_n]^T,$

$$\mathbf{A_a} = \begin{bmatrix} -a_1(1-k_1)-k_2 & 1 & 0 \ldots & 0 \\ -a_2(1-k_1)-k_3 & 0 & 1 \ldots & 0 \\ \vdots & & \ddots & \vdots \\ -a_{n-1}(1-k_1)-k_n & & 0 & 1 \\ -a_n(1-k_1) & & & 0 \end{bmatrix}.$$

Thus (14.2.13) is rendered the dual of the covariance time-update (13.7.3) of the state-space formulation of the optimal LQ control problem; hence the vector implementation using U-D factorization (Appendix 13C) can be directly used, where $\mathbf{A_a}$ and f are substituted for $\mathbf{A}$ and q (see (13C.2a)) respectively in the FORTRAN implementation. The rest of the smoothing problem is straightforward.

(c) Data storage requirements

The present algorithm requires storing of the n-vector $\mathbf{k}(k)$ and the scalars $f(k)$ and $\tilde{y}(k|k+1)$ for $k = 0,1,\ldots,N$, are produced from the forward filtering. In addition, $\hat{\mathbf{x}}(k|k-1)$ and $\mathbf{P}(k|k-1)$ need to be stored only for specific values of k for which the smoothed estimate $\hat{\mathbf{x}}(k|N)$ is to be computed. Thus the storage requirement is much less compared with the RTS algorithm, which requires $\hat{\mathbf{x}}(k|k)$, $\hat{\mathbf{x}}(k+1|k)$, $\mathbf{P}(K|k)$ and $\mathbf{P}(k+1|k)$ to be stored for each value of k.

(d) Stability

The present algorithm is numerically stable because (i) no matrix inversions are involved and (ii) the smoothing recursions are computed using the adjoint variables rather than the smoothing error covariances $\mathbf{P}(k|N)$.

In fact, $\mathbf{P}(k|N)$ need not be computed at all except for diagnostic purpose.

14.2.2 Fixed-point Smoothing

Often, the initial conditions or conditions at a specific time point is of prime concern as the process or the experiment progresses with time, e.g. the time of firing of a rocket, when a satellite is tracked from a ground-station. In such cases, the objective is to estimate $\hat{x}(k|j)$, where $j>k$, which is given by the following algorithm.

$$\hat{x}(k|j) = \hat{x}(k|j-1) + g(k|j)\tilde{y}(j|j-1), \tag{14.2.16}$$

$$g(k|j) = S(k|j)c(j)f(j), \tag{14.2.17}$$

$$S(k|j) = S(k|j-1)A_a^T(j|j-1), \tag{14.2.18}$$

for $j = k+1, k+2, \ldots,$ where nxn matrix A_a and scalar f are defined as

$$A_a(j|j-1) = A(j|j-1)\ [I - k(j-1)c^T(j-1)],$$

$$f(j) = (c^T(j)P(j|j-1)c(j) + R(j))^{-1},$$

and

$$\tilde{y}(j|j-1) = y(j) - c^T(j)\hat{x}(j|j-1).$$

The iterations are initialized with

$$\hat{x}(k|j-1) = \hat{x}(k|k), \quad \text{and}$$

$$S(k|j) = S(k|k+1) = P(k|k)A^T(k+1|k),$$

in (14.2.16) and (14.2.17) respectively.

If necessary, $P(k|j)$, the covariance of the fixed-point smoothing error, $\tilde{x}(k|j) = x(k)-\hat{x}(k|j)$, may be computed from

$$P(k|j) = P(k|j-1) - g(k|j)f^{-1}(k)g^T(k|j), \tag{14.2.19}$$

with the initialization, $P(k|j-1) = P(k|k)$.

Summary

(1) Perform forward filtering (14.2.3 - 14.2.6) over the available data up to the specified point in time k.

(2) From the point k onwards, for each time-step, perform smoothing iterations (14.2.16 - 14.2.18), following the forward filtering to compute $\hat{x}(k|j)$, where $j=k+1$, $k+2$, etc.

(3) Computation of error covariance $P(k|j)$ is optional.

Remark: The algorithm discussed here is forward recursive in time, and hence can be performed on-line. The smoother does not require prior information of the final measurement.

14.2.3 Fixed-lag Smoothing

The fixed-lag smoothing problem can be configured as a fixed-interval smoothing problem, the objective being to compute the smoothed estimate $\hat{x}(k|k+L)$ of $\hat{x}(k)$, after a delay of L time points, where L is fixed. The implementation involves the following steps:

(1) Compute forward filtering (14.2.3 - 14.2.6) up to the time point (k+L).

(2) Perform backward recursions of augmented variables λ and Λ in (14.2.12 - 14.2.13) to determine $\lambda(k|k+L)$ and $\Lambda(k|k+L)$, the initial conditions being $\lambda(k+L+1|k+L) = 0$ and $\Lambda(k+L+1|k+L) = 0$.

(3) Compute fixed-lag smoothed estimate $\hat{x}(k|k+L)$ from (14.2.10).

For further details, refer to Sec.14.2.1.

14.2.4 Observations and Comparative Study

(a) *Relationship with Optimal Filter*

The Kalman filter is an integral part of the optimal smoother, which is also linear and has the same dimension as the Kalman filter. The following comparative features are worth noting.

(1) Like the optimal filter, all the presented smoothing algorithms are driven from the innovations process

$$\tilde{y}(k|k-1) = y(k) - c^T(k)\hat{x}(k|k-1).$$

(2) As in the case of the optimal filter, the time points k, k-1,... etc. need not be equispaced for the optimal smoother.

(3) Unlike the optimal filter, the error covariance $P(k|k+i)$, are not an integral part of the smoothing iterations and hence need not be computed; they can be used for diagnostic purposes.

(4) The stochastic error processes for all three categories of optimal smoothing are *Gauss-Markov-2* processes, unlike optimal filtering and prediction cases which are *Gauss-Markov* processes (Sec.2.2.2).

(b) *Improvements due to smoothing*

(1) By definition, the optimally smoothed estimate produces

minimum smoothing error covariance **P**(k|N). So a comparison between the diagonal elements of **P**(k|N) and the filter covariance **P**(k|k), gives the first hand information about the improvement in the state estimates due to smoothing.

(2) The improvement due to smoothing is realizable when there is considerable driving noise w and measurement noise v.

(3) The improvement due to smoothing, given by (**P**(k|N)-**P**(k|k)) increases with the increase in N. In fact, the rate of increase depends on the dominant time constant of the forward Kalman filter which is given by the eigenvalues of $(\mathbf{A} - \mathbf{k}\mathbf{c}^T)$. In general the smoothing is most effective over the region, two to three times the dominant time constant of the Kalman filter from k.

(4) For a limited horizon, **P**(k|N) is relatively higher for values of k close to N which is due to the transient state of the backward filter.

(c) *Optimality*

The optimality of optimal smoothing is confined to the condition of satisfying the minimum mean square estimation error criterion (for example (14.2.1)). However it does not necessarily guarantee minimum phase shifts or time-delay consequent to smoothing, which is important particularly in prediction studies.

(d) *Miscellaneous features*

(1) The three broad categories of optimal smoothing are mutually equivalent (i.e. one can be derived from the others), although they concern different ways of configuring the optimal smoothing problem.

(2) For short data sets, fixed-interval smoothing is most appropriate.

14.3 Bidirectional Filtering

Two fundamental issues related to smoothing or filtering are
(i) the mean square estimation error and
(ii) the real-timeliness of the estimates.
Often it is necessary to perform smoothing or filtering of the data before use in the predictor. This is to separate

high frequency noise from the data. However, smoothing or filtering should not damage the real-timeliness of the data (for example, it should not introduce lag or time delay in the data), as otherwise the prediction performance deteriorates. In this section, first some basic off-line approaches to time-lag-free smoothing are discussed. One problem with the off-line method is the inability to process the terminal data. So the concept of real-time bidirectional filtering is introduced where a real-time predictor is used in combination with the filter.

14.3.1 Off-line Method

Here, frequency domain low-pass filtering is performed bidirectionally. In other words, the data are passed through the same filter in opposite directions to compensate for the time-delay and finally delay-free estimates are produced.

The implementation of the filter is detailed in Fig.14.3.1(a); it involves the following steps:

(a) a forward pass of the data through the chosen filter,

(b) reversal in time of the filtered data obtained from (a),

(c) a forward pass of the filtered data obtained from (b) through the same filter as in (a).

(d) reversal in time of the filtered data so obtained, which are the phase-shift free smoothed estimates.

An alternative but equivalent scheme is also presented in Fig.14.3.1(b).

Design and analysis

The design of the bidirectional filter involves two main issues:

(1) The extent of filtering, that is the highest frequency to be permitted in the filtered data is a prime consideration. If a first order filter given by

$$F(q^{-1}) = \frac{1-\alpha}{1-\alpha q^{-1}}, \qquad 0<\alpha<1. \qquad (14.3.1)$$

is used, a slow pole (that is a higher value of α) will have a lower cut-off frequency (that is relatively lower frequencies will be passed as discussed in Appendix 14A).

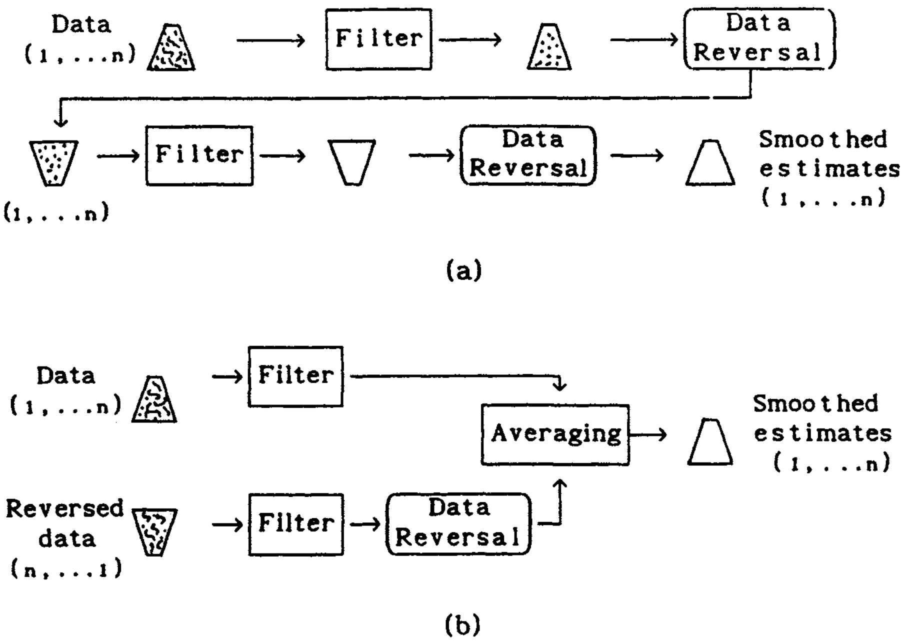


Figure 14.3.1 Bidirectional filter schemes:
(a) One data reversal following each of the two passes through the same filter produces filtered data;
(b) Same lag or phase-shift in opposite direction produced by single pass of the data and the reversed data through two similar filters; the filtered data combine in phase to produce the final output.

As too fast a pole will not be able to result in desirable elimination of high frequency components, too slow a pole is likely to eliminate useful information in the data. So appropriate design of the filter is important.
(2) The final estimates obtained through bidirectional filtering will ideally have zero time-lag, if the characteristics of the forward and the reverse filters are identical.

Remarks
(1) Longini *et al* (1975) describe a hardware implementation of bidirectional filtering. The electronically taped data are passed through a filter and are retaped. The tape

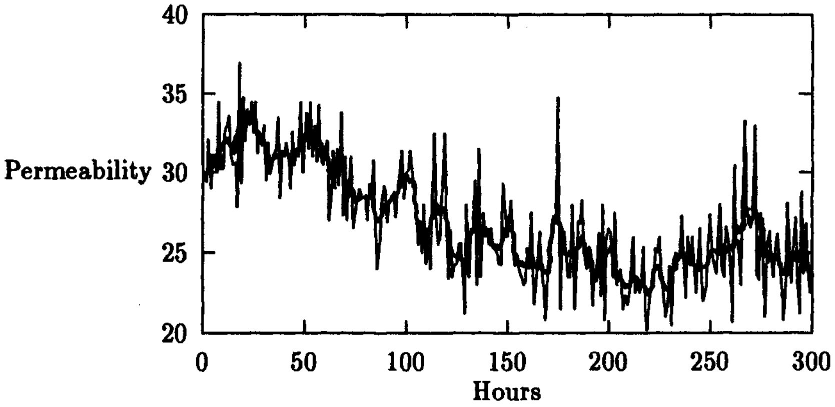


Figure 14.3.2 Bidirectional filtering of noisy green-mix permeability data with $\alpha = 0.6$ in (14.3.1).

containing the filtered (retaped) data is played in (physically) reversed direction through the same filter; the consequent output is taped, which when run in reverse direction produces time-delay-free filtered data.

(2) The *bidirectional* feature is implicit with the state-space formulation of the optimal smoothing methods incorporating a forward filter and a reverse filter.

(3) The centred moving averaging (Sec.4.2.1 and Appendix 4),is also a form of bidirectional filtering.

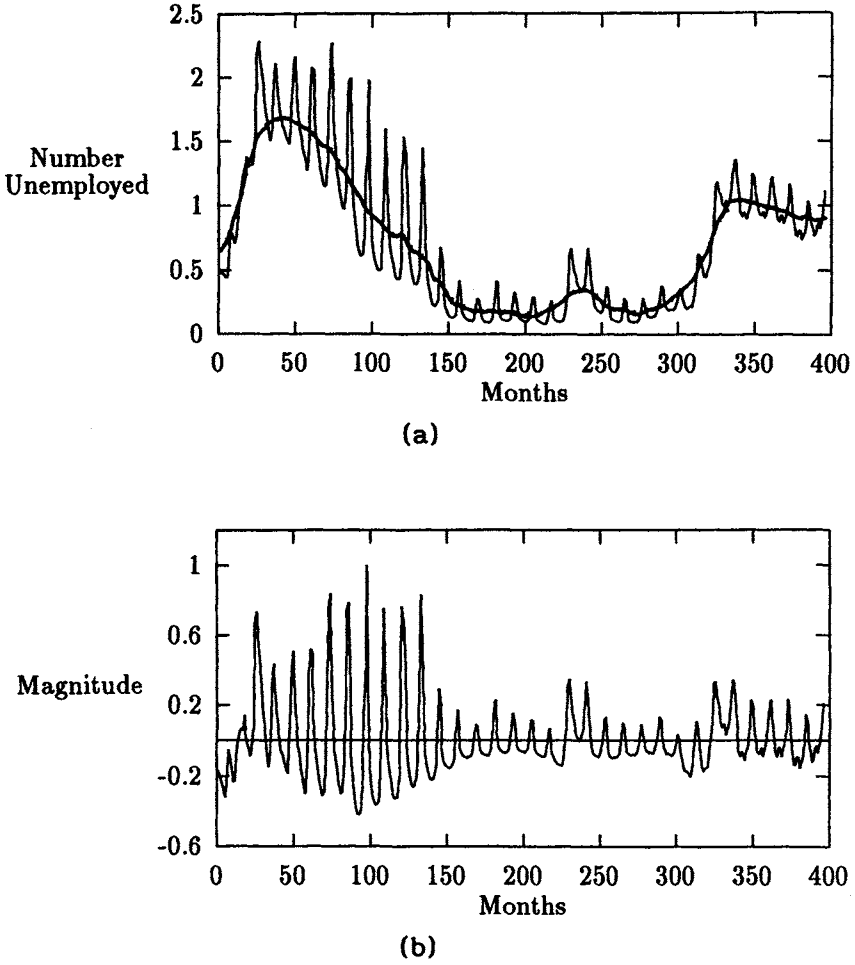
Example 14.3.1 Filter the green-mix permeability data.

The permeability of green-mix is an important process variable in the iron-ore sintering process (discussed in Sec. 5.6.1). The series (see Appendix 14B) contains 2-minutely noisy data for 5 hours from an iron and steel plant.

A first order filter (14.3.1) is used with $\alpha = 0.6$. The results of bidirectional filtering are presented in Fig. 14.3.2; the delay free output of the bidirectional filter is shown by the bold line.

Example 14.3.2 Determine the trend in the German unemployment series using bidirectional filtering.

The trend of a time series is basically the slowly varying



(a)

(b)

Figure 14.3.3 (a) Trend (———) estimation of the German unemployment series (Appendix 7E) with $\alpha = 0.9$. (b) Trend extracted German unemployment series.

trend component, which can be separated by low-pass filtering with sufficiently low cut-off.

The present series (Fig.14.3.3.(a)) is a monthly data sequence with yearly periodicity. A bidirectional filter with an α=0.9 is chosen. The trend extracted residual series is shown in Fig.14.3.3(b).

The sensible choice for α requires the consideration of the lowest frequency component that can be present due to the yearly periodicity; the cut-off point of the filter has to be chosen lower (see Sec.2.5.4).

14.3.2 Real-time Filtering

The term *real-time* signifies *relevance to time*. The objective of real-time filtering is to filter the current data with minimum time lag, to be available for immediate use.

It is known that the smoothing characteristics of conventional smoothers including the optimal smoothers deteriorate considerably towards the terminal points. In conventional unidirectional filtering, including Kalman filtering, the filtered estimates suffer from considerable phase-shift. Ideally, the phase-shift or lag should be zero, particularly if the filtered values are to be used by a predictor. Real-time filtering with minimum delay can be achieved by trying to imitate the bidirectional filtering process on the extrapolated data set. In other words, the available data are used for multistep prediction (say, $\hat{y}(k+1|k),\ldots,\hat{y}(k+p|k)$) and then the bidirectional filtering is performed on the data (say, $y(0)$, $y(1),\ldots,\hat{y}(k+p|k)$) to produce the desired estimate $\hat{y}(k|k)$. The consequent minimization of lag or bias in the data will be largely dependent on the quality of prediction.

Real-time filtering of the data has been used for sinter quality prediction discussed in Secs.5.6.2 and 5.6.3.

14.4 SMOOTHING AND FILTERING USING ORTHOGONAL TRANSFORMATION

Introduction

As discussed in Chapter 7, orthogonal transformation results in compaction and relative decorrelation of information in the transformed components of the data. Consequently separation of unwanted transformed components from the rest is easier. Usually the contaminations in the data, which are uncharacteristic of the underlying process, are confined to transformed components with low energy content or low signal strength, which are eliminated. Orthogonal transformation being a linear and reversible operation, data smoothing

involves three basic steps:
(1) orthogonal transformation of the data,
(2) the elimination of the unwanted or insignificant transformed components, and
(3) the data reconstruction through reverse transformation.
The intermediate step of elimination of unwanted components causes actual smoothing. Besides smoothing, selective filtering is also possible, where the signal of interest may or may not be the prime component.

The degree of concentration of information in the transformed components and the degree of decorrelation among different components will determine how efficiently smoothing and filtering can be performed. The numerical robustness properties of the orthogonal decomposition will also be associated with the smoother or the filter.

The present studies are limited to the use of singular value decomposition for smoothing and filtering, although other types of orthogonal transformations can also be used.

Comparison with optimal smoothers

As against state-space based optimum smoothers discussed in Sec.14.2, the following comparative features of orthogonal transformation based approaches can be observed.

(a) The orthogonal transformation based smoother has structural similarity with the fixed interval smoother, the smoothing error covariance matrix $\mathbf{P}(k|N)$ can be used as a diagnostic measure, whereas in the case of the orthogonal transformation smoothers the energy contained in the signal and that in the eliminated components can provide a measure of the relative strength of the noise.
(b) One fundamental structural difference between the state-space smoother and the orthogonal transformation smoother is that the former assumes causality, whereas the latter does not; it is mainly based on the eigen properties. Hence while the noise statistics have to be prespecified in the case of state-space smoothers, it is not required for the orthogonal transformation based smoother.
(c) For state-space smoothers the smoothing property deteriorates towards the end of the data sequence and the error covariance tends to increase. In the case of the orthogonal transformation smoother, the smoothing depends on the total information contained in the data set concerned,

and the smoothing performance is not expected to be different towards the terminals; the indifference to causality is another way of explaining the same thing.

Remarks

(a) Since Fourier transformation decomposes data into frequency components, the frequency domain filtering is equivalent to the separation of components after Fourier transformation.

(b) Different classes of orthogonal transformation result in different types of decompositions, and the corresponding subsequent smoothing or filtering algorithm will have its own characteristic features. For example, the singular value decomposition (SVD) based filtering can be independent of the frequency distributions of the data, and hence can be quite different from Fourier transformation based filtering; in the case of the latter the orthogonality is in the frequency domain, whereas in the case of SVD the orthogonality is in the information space corresponding to the algebraic configuration of the data matrix.

14.5 SMOOTHING AND FILTERING USING SVD

Applications of singular value decomposition to nearly repetitive periodic processes is considered, where the pattern, the length of period and the average value over a period may all change to a limited extent. The three specific applications studied are smoothing, pattern estimation and selective extraction of pattern through filtering.

Singular value decomposition has been discussed in detail in Sec.7.6. SVD of a real mxn matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T;$$

if min(m,n) = p, and if $\mathbf{A}$ is of rank r (≤p), then

$$\mathbf{A} = \sum_{i=1}^{r} \mathbf{u}_i s_i \mathbf{v}_i^T, \tag{14.5.1}$$

where s_i are the singular values: $s_1 \geq s_2 \geq \ldots \geq s_r > 0$, and $s_{r+1} = s_{r+2} = \ldots = s_p = 0$. $\mathbf{u}_i s_i \mathbf{v}_i^T$ and $\mathbf{u}_j s_j \mathbf{v}_j^T$ are orthogonal to each other for i≠j, where i and j are between 1 and r.

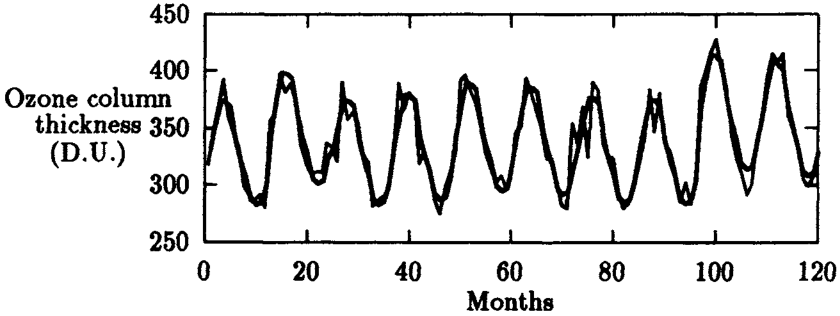


Figure 14.5.1 The smoothing of the atmospheric ozone column series.

14.5.1 Smoothing

In the present context, smoothing is obtained through successive rejection of orthogonal components starting from the end having lowest energy. In (14.5.1), the component having the lowest energy is given by $u_r s_r v_r^T$. If $\bar{S}$ is the diagonal matrix of g singular values after elimination of (r−g) smallest singular values, the corresponding truncated singular vector matrices being $\bar{U}$ and $\bar{V}$, the *smoothed* data matrix will be given by

$$\hat{A} = \bar{U}\bar{S}\bar{V}^T,$$

where $\bar{U}$, $\bar{S}$ and $\bar{V}$, will be mxg, gxg, gxn matrices respectively. There is no definite rule regarding how many modes connected with the smaller singular values should be eliminated. Usually it is based on the relative strength with respect to the prime mode $u_1 s_1 v_1^T$, which is given by the ratio s_i^2/s_1^2 for the i-th decomposition component.

An application of SVD for data smoothing follows.

Example 14.5.1 Smoothing of monthly series of ozone column thickness

The (monthly data on the) thickness of the ozone column shows yearly periodicity with a large amount of variability. The data are given in Appendix 7B. As shown in Fig.14.5.1, the series is periodic with a lot of variability between the periods. For this study, data for 21 years (1932 to 1952) are used.

The data are arranged into the rows of a 21x12 matrix **A**, with the data for each year arranged row-wise. The singular values of **A** are obtained as follows:

s_1 to s_{12}: 5390.1, 121.9, 86.4, 79.3, 67.2, 56.7,
49.8, 41.1, 34.9, 32.4, 22.4, 19.3.

The smoothed $\hat{\mathbf{A}}$ is constructed by considering only the most dominant singular value (i.e. g = 1). The smoothed series is reconstruced from $\hat{\mathbf{A}}$ and is as shown in Fig.14.5.1.

Remark: This scheme of smoothing may be applied for the reconstruction of the missing data. Smoothing is performed with interpolated data in place of the missing data; the corresponding smoothed estimates are the reconstructed data.

14.5.2 Pattern Estimation

The problem addressed here is the identification of a nearly repetitive pattern from multiple sets of data from the same experiment or process, using SVD. Here, the term *pattern* is used to mean the distribution of the data series over a period. For example (a) stress tests on blocks of the same metal will show almost similar patterns if a number of experiments are conducted, or (b) the power load demand on a substation for a particular day of the week will show similar patterns for a number of consecutive weeks. Such processes can be characterized by the following features:

(a) Repetitiveness of the process.

(b) Periodic or nearly periodic nature, i.e. the period length may be same, or it may vary to some extent. For example the number of measurements recorded for different experiments may not be exactly the same when repeated.

(c) Absence of any conspicuous trend. In cases where there is a trend, it has to be removed before pattern estimation.

The objective is to estimate *one* pattern spanning one period from the available periodic or nearly periodic sets of data.

The characterization of periodic and nearly periodic processes through singular value decomposition is discussed in Sec.7.7.1, which is used for pattern estimation in the present case.

Problem formulation

It is assumed that the data sets concerned are nearly periodic. The period length may be fixed or may vary to some extent. In the latter case, the optimum period length is determined using the SVR spectrum of the data series formed from the available successive periods; the periods, having period length different from the optimum, are stretched or compressed (as is necessary) uniformly to the optimum period length as discussed in Sec.11.2.1. A simplified alternative is to use the maximum period length in the data as n and to use interpolated data wherever necessary to make periods equal in length.

The m consecutive periods of the data are arranged in an mxn matrix **A** so that the consecutive periods are aligned into the consecutive rows of **A**; n is the period length. Any missing data may be replaced with interpolated data.

It is known (Sec.7.7.1) that a matrix with nearly repetitive rows will produce one prime singular value on singular value decomposition (SVD), with the other singular values being insignificantly small, i.e. $s_1 \gg s_2$ in S:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \qquad \mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \ldots \mathbf{u}_m],$$
$$\simeq \bar{\mathbf{A}} = \mathbf{u}_1 s_1 \mathbf{v}_1^T, \qquad \mathbf{u}_1 = [u_{11}\ u_{21}\ \ldots\ u_{m1}]^T,$$

where $\mathbf{u}_1$ and $\mathbf{v}_1$ are the first columns of U and V respectively. Thus **A** is approximated by a rank-1 matrix $\bar{\mathbf{A}}$. If u_{i1} is the i-th element of $\mathbf{u}_1$, the i-th row of $\bar{\mathbf{A}}$ is given by $u_{i1}s_1\mathbf{v}_1^T$. $\bar{\mathbf{A}}$ thus represents the principal periodic pattern in **A**. The points to note are as follows.

(a) the rows of $\bar{\mathbf{A}}$ will contain repeating periods or patterns, there being m, n-long periods,

(b) $\mathbf{v}_1^T$ represents the basic pattern of the periods, which is the same for all the periods,

(c) the elements of $\mathbf{u}_1 s_1$ represent the scaling between the rows, the i-th row of $\bar{\mathbf{A}}$ being weighted by $u_{i1}s_1$.

The objective is to estimate the average row-energy, periodic pattern of $\bar{\mathbf{A}}$:

$$\hat{\mathbf{y}}_{av} = \hat{u}\hat{s}\hat{v}^T.$$

Average energy pattern

The energy contained in the i-th row of **A** is given by

$$\begin{aligned} E_i &= [u_{i1}s_1\mathbf{v}_1^T]^T[u_{i1}s_1\mathbf{v}_1^T]. \\ &= u_{i1}^2[s_1\mathbf{v}_1^T]^T[s_1\mathbf{v}_1^T]. \end{aligned}$$

Hence the average energy contained in a row of $\bar{\mathbf{A}}$ is given by

$$\begin{aligned} E_{av} &= \frac{1}{m}\sum_{i=1}^{m} E_i, \\ &= [\frac{1}{\sqrt{m}}\, s_1\mathbf{v}_1^T]^T[\frac{1}{\sqrt{m}}\, s_1\mathbf{v}_1^T] = \frac{s_1^2}{m}, \end{aligned}$$

since $\mathbf{u}_1^T\mathbf{u}_1 = 1$, and $\mathbf{v}_1^T\mathbf{v}_1 = 1$. Hence the desired average energy periodic pattern is given by

$$\hat{\mathbf{y}} = \frac{s_1\mathbf{v}_1^T}{\sqrt{m}}. \qquad (14.5.1)$$

Remarks

(a) The present method is conceptually different from *time averaging*. Here, the pattern is extracted through the separation of the noise, which is eliminated as an algebraically orthogonal component to the pattern of interest. Simple Averaging being eqivalent to low(frequency)-pass FIR filtering, any noise which is orthogonal to the signal cannot be separated by averaging, if the frequency components of the noise are also present in the signal.

(b) Besides pattern estimation, one direct application of the proposed method is data compression. For example, if one is interested in long-term storage of the ECG data, only one element (s_1) and one vector ($\mathbf{v}_1$) will be required to be stored to represent the ECG pattern over a short span of time.

(c) Where there are more than one (say, g) singular values dominant, the estimated pattern will be given by

$$\hat{\mathbf{y}} = \sum_{i=1}^{g}\frac{s_i\mathbf{v}_i^T}{\sqrt{m}}, \qquad (14.5.2)$$

if the elements of $\mathbf{u}_i$ do not change sign.

Example 14.5.2 Estimation of the daily electric power load pattern on a substation

The hourly data on the electric power load on a substation

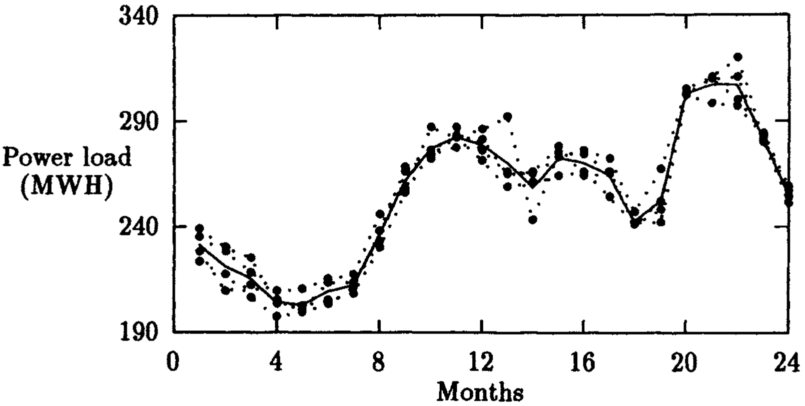


Figure 14.5.2 The Monday's load pattern of electrical power over a period of 24 hours.
. . . . . data patterns, ———— estimated pattern.

(Apppendix 7D) shows a daily periodicity as well as weekly and yearly periodicities. Consider the problem of estimating the Monday-load-pattern over a month.

The data for the four consecutive Mondays are arranged into a 4x12 matrix **A**, and the load pattern is estimated using (14.5.1). Here, only the most dominant decomposition component was used for pattern estimation.

14.5.3 Selective Filtering

In a data series or signal, besides the noise, there may be two or more information components; the signal component of interest may not be the most dominant SV-decomposed component. In other words, the signal component of interest may be mixed with stronger unwanted auxiliary signals. Selective filtering concerns separation of a signal component by selective elimination of unwanted decomposition components.

In the case of a periodic signal arranged into a matrix, if the signal components are mutually orthogonal, they will be associated with specific singular values s_i and hence with the corresponding SV-decomposition components $u_i s_i v_i^T$, where

$$A = USV^T = \sum_{i=1}^{p} u_i s_i v_i^T.$$

The selective separation of the components $u_i s_i v_i^T$ can be performed directly in such cases.

In practice, signal components are rarely orthogonal to each other, so the following features of a data matrix decomposed by SVD are to be noted.

(a) A predominant first singular value is indicative of the presence of a strong periodic component given by $u_1 s_1 v_1^T$.

(b) If the last few singular values are insignificantly small, they are associated with the noise components.

(c) The singular values which are in between may contain useful information.

A case study illustrating the application of selective filtering follows.

14.5.4 Case Study: Fetal ECG Extraction from Maternal ECG

The ECG signal recorded from the abdominal lead on the mother shows a fetal ECG component submerged within the maternal ECG. From a clinical point of view fetal ECG contains useful information about fetal health. Besides the interfering maternal ECG signal, there is noise due to maternal muscle contractions, motion artifacts due to the movements of the baby and the mother etc. Thus extraction of the fetal ECG signal can be a difficult problem.

Widrow *et al* (1975) use the method of adaptive noise cancellation to produce the fetal ECG signal component. The mother's ECG signal obtained from the chest lead is used to cancel the maternal ECG signal component from the composite ECG signal obtained from the abdominal lead of the mother. Another possible approach is to deduce fetal ECG signal through weighted summation of a number of ECG signals obtained from suitably positioned electrodes on the mother (Callaerts *et al*, 1990).

The method discussed here uses only *one* signal, that is the maternal abdominal ECG signal, from which the fetal ECG component is extracted through selective filtering.

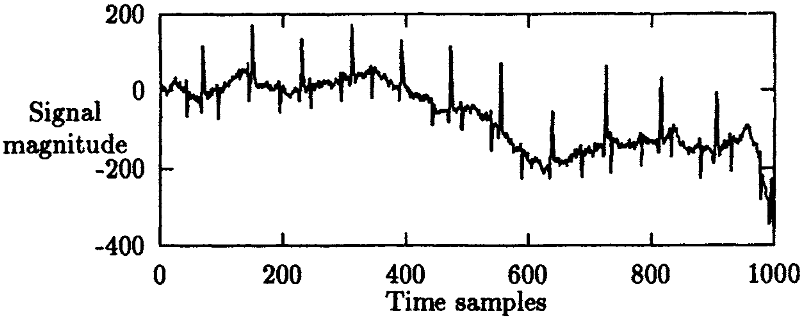


Figure 14.5.3 Maternal ECG (composite) signal from the abdominal lead of the mother, showing maternal and fetal ECG components.

Basic principle

The maternal abdominal ECG signal (MAECG) is composed of the maternal ECG (MECG) signal, the fetal ECG (FECG) signal, the high frequency noise components and the trend component due to low frequency offsets and drifts. The objective is to extract a reasonably clean FECG signal from the composite MAECG signal.

Note that (see Fig.14.5.3) the MECG and FECG components are nearly periodic, with marginal variations in the periodic patterns and period lengths; the MECG and FECG are usually asynchronous with each other. The FECG can be extracted as follows.

(1) First, the composite MAECG signal is bidirectionally filtered (Sec.14.3.1) to separate the low frequency drift component from the data.

(2) The optimum period length n_M for the maternal ECG component in the composite MAECG signal is determined by inspection or by SVR spectrum (Appendix 11). In the case of the former, the maximum period length may be chosen. For the SVR spectrum, the length corresponding to the highest peak is chosen.

(3) The composite MAECG data are arranged into a matrix **A** with a row length n_M (see the following Remark). Each maternal ECG period occupies a row of **A** with the peak

excursions lying in the same column. SVD is performed on **A**: $\mathbf{A} = \mathbf{U}_A\mathbf{S}_A\mathbf{V}_A^T$; the most dominant decomposition component of **A**, $\mathbf{A}_M = \mathbf{u}_{A1}s_{A1}\mathbf{v}_{A1}^T$, represents the maternal ECG signal. The fetal ECG signal remains contained in the residual, $\mathbf{A}_R = \mathbf{A}-\mathbf{A}_M$.

(4) The data sequence $\{y_R(.)\}$ is formed from the consecutive rows of $\mathbf{A}_R$. The most dominant nearly periodic component in $\{y_R(.)\}$ will be the fetal ECG component. The period length n_F of the fetal ECG component is determined through inspection or by SVR spectrum analysis of $\{y_R(.)\}$.

(5) $\{y_R(.)\}$ is arranged into a matrix **B** with the row length equal to n_F. SVD is performed on **B**: $\mathbf{B} = \mathbf{U}_B\mathbf{S}_B\mathbf{V}_B^T$. The most dominant SV-decomposed component of **B**, $\mathbf{B}_F = \mathbf{u}_{B1}s_{B1}\mathbf{v}_{B1}^T$, is the extracted FECG component (the consecutive rows of $\mathbf{B}_F$ contain the consecutive periods of FECG).

Remarks

(a) Due to physiological reasons the period lengths of both the MECG as well as the FECG may vary to certain extent. To accommodate each period in a row of the matrix **A** or **B** in step (3) above, two approaches may be followed.

(i) A period may be stretched to the chosen period length by equivalent interpolation or extrapolation using (11.2.1). This operation is required at the time of forming the data matrix. The reverse operation is done at the time of forming the data sequence from the separated matrix (like $\mathbf{A}_M$) or the residual matrix (like $\mathbf{A}_R$).

(ii) If the maximum period length is chosen as the row length, when the row falls short of data at the teminals of a row, interpolated values may be used. The reverse operation is performed while forming the data sequence from the matrix.

(b) In the formation of both **A** and **B**, it is necessary to ensure that the peaks of the prime ECG signals lie in the same column (or the same few columns) in the different rows of the matrix. It is also important that the operation of interpolation or extrapolation or padding applied in forming the matrix is reversed at the time of forming the extracted data sequence or the residual data sequence from the respective matrices.

(c) There is no frequency domain method of filtering which is equivalent to the SVD based selective filtering applied here.

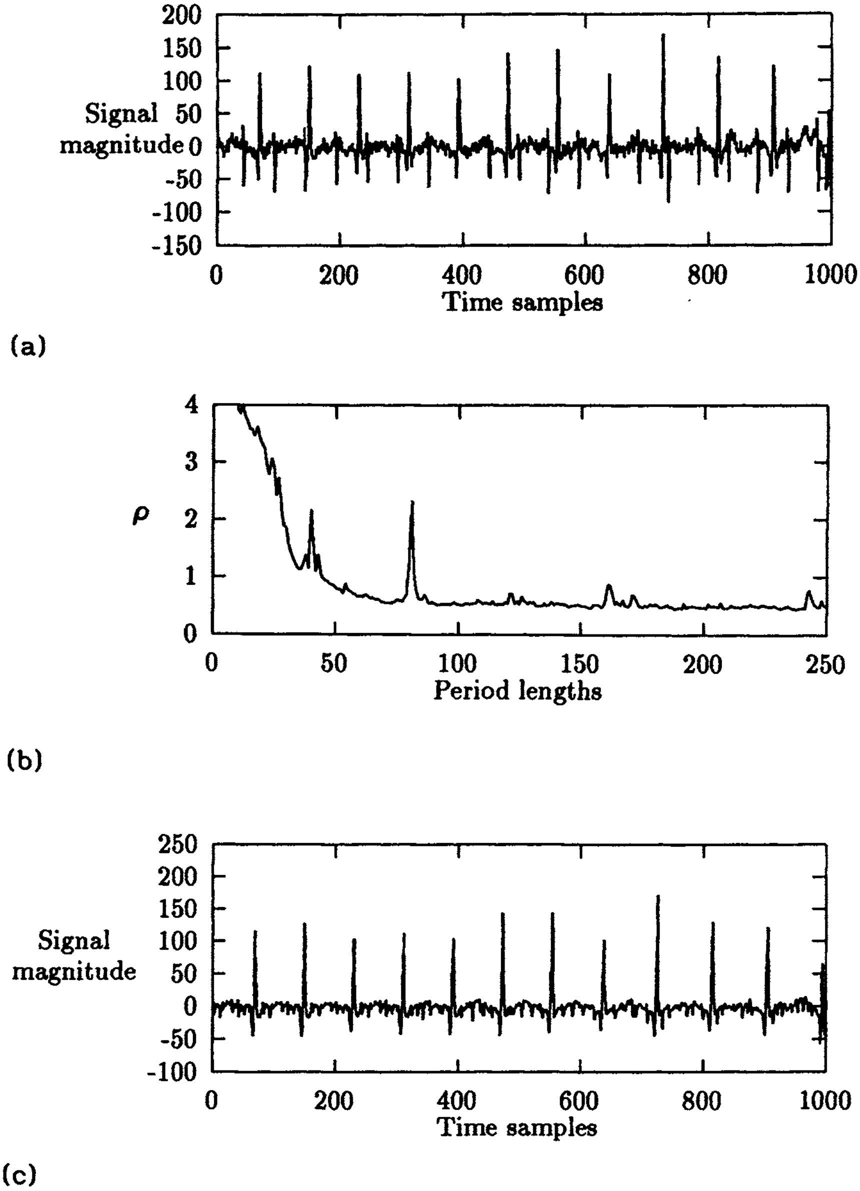


Figure 14.5.4 (a) The trend removed composite maternal ECG signal shown in Fig.14.5.3.
(b) The SVR spectrum of composite ECG (MAECG) signal,
(c) The extracted maternal ECG (MECG) signal.

Example 14.5.4 Fetal ECG extraction using real-life data

The data used in this example (Appendix 14C) were obtained from a mother with a gestation period of 37 weeeks. As shown in Fig.14.5.3, the series shows the fetal ECG signal mixed with the maternal ECG signal and riding on an interfering low frequency trend component.

To remove the low frequency trend in the composite maternal abdominal ECG signal, it was bidirectionally filtered with the filter pole at 0.88 (i.e. $\alpha = 0.88$ in (14.3.1)). The filtered signal $\{y(.)\}$ is shown in Fig.14.5.4(a). The SVR spectrum (Fig.14.5.4(b)) on $\{y(.)\}$ showed a peak at a period length of 81; the data showed a variation of the period length from 79 to 90. So wherever necessary, the periods were uniformly extrapolated or compressed (using (11.2.1)) and a 12x81 matrix **A** was formed. **A** was SV-decomposed and $\mathbf{A}_M$ representing the maternal ECG component (Fig.14.5.4(c)) was extracted.

The SVR spectrum on the residual data sequence showed peaks at period lengths of 25, 49 and 98 (Fig. 14.5.5(a),(b)). Since there was no peak at 75, n_F was chosen as 49, and 20x49 matrix **B** was formed. **B** was SV-decomposed and the fetal ECG component $\mathbf{B}_F$ was extracted as shown in Fig.14.5.5(c).

14.6 CONCLUSIONS

Some methods for separation or extraction of usable information from the available data have been presented. First, optimum smoothing in state-space framework was discussed, which was to familiarize the reader with the various issues connected with smoothing. This was followed by studies on bidirectional filtering, which can be used to perform smoothing with minimum lag or phase shift. The bidirectional processing is a characteristic feature which is incorporated in many algorithms for bias-free processing; for example the fixed interval smoother uses similar forward-backward passes, and the centred moving average (Sec.4.2.3 and Appendix 4) used in the time series analysis is also effectively similar in concept.

Orthogonal transformation offers a numerically robust method of smoothing and signal extraction. The smoothing is performed through the elimination of insignificant

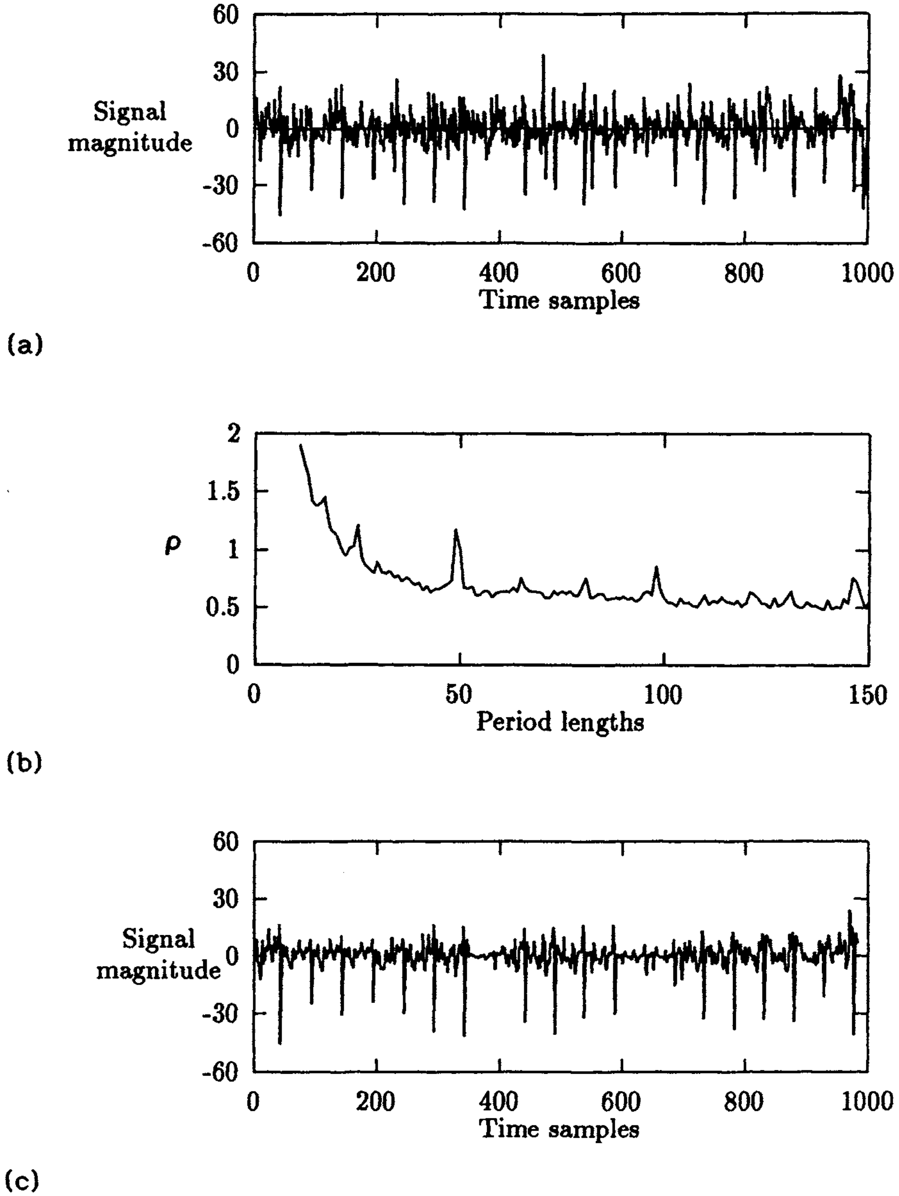


Figure 14.5.5 (a) The residual series after removal of the maternal ECG component.
(b) The SVR spectrum of the residual series,
(c) The extracted FECG signal.

on singular value decomposition (SVD). Besides smoothing, the potential of the SVD based methods in signal extraction and pattern estimation in a noisy environment was also demonstrated through application studies. The approach depends on the repetitive nature of the signal component of interest, and hence the data are appropriately configured for analysis. A case study on fetal ECG extraction from maternal ECG showed that extraction is possible with only one signal (i.e. the maternal ECG signal from the abdominal lead), and irrespective of low signal to noise ratio; the other available methods of fetal ECG extraction require one or more additional signals. The application of orthogonal transformation for smoothing and filtering is an area of active research, and the present study has been only a glimpse of its enormous potential.

REFERENCES

Remarks: Optimum smoothing is a widely studied and reported topic; some of the selected works are listed here. Some standard texts covering various aspects of smoothing are [1,5,13,14,15]. Fixed interval smoothing is also studied in [2,3,7,16,18]. An early implementation of bidirectional filtering appeared in [12]; the related topic of centred moving average featured in Appendix 4. Real-time bidirectional filtering features in [9], and also in Sec.5.6.

Orthogonal transformation has been introduced in Chapter 7. The concept of the use of SVD for smoothing and selective filtering is discussed in Secs.7.6, 7.7, 11.2 and 11.4. A study on ECG data compression appears in [8]. Fetal ECG extraction from maternal ECG has been an area of active interest. An early work appeared in [19]; methods based on SVD are presented in [6,10,11,17].

[1] Anderson, B.D.O., and J.B. Moore (1979): *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, N.J.

[2] Bierman, G.J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[3] Bierman, G.J. (1983): 'A new computationally efficient fixed-interval discrete-time smoother', *Automatica*, 19(5), 503-511.

[4] Bryson, A.E., and Frazier, M. (1963): 'Smoothing for linear and nonlinear dynamic systems', *Proc. Optimum Systems Synthesis Conf.*, U. S. Air Force Tech Rep. ASD-TDR-063-119.

[5] Bryson, A.E., and Y.C. Ho (1969) *Applied Optional Control*, Ginn(Blaisdell), Walthlam, Mass.

[6] Callaerts, D., B. De Moor, J. Vandewalle, and W. Sansen (1990): 'Comparison of SVD methods to extract the foetal electrocardiogram from cutaneous electrode signals', *Medical and Biological Engineering and Computing*, **28**, 217-224.

[7] Fraser, D.C., and J.E. Potter (1969): 'The optimum linear smoother as a combination of two optimum linear filters', *IEEE Trans. Autom. Control*, **AC-14**, 387-390.

[8] Jalaleddine, S.M.S., C.G. Hutchens, R.D. Strattan, and W.A. Coberley (1990): 'ECG data compression techniques - a unified approach', *IEEE Trans. on Biomedical Eng.*, 37(4), 329-343.

[9] Kanjilal, P.P., and E. Rose (1986): 'Application of multivariable self-tuning predictor', *Proc. of IEEE*, 74 (2), 377-379.

[10] Kanjilal, P.P., S. Palit, and P.K. Dey (1993): 'Fetal ECG extraction from maternal ECG and data compression using singular value decomposition', Proc. of 12th IFAC World Congress, Sydney, 449-452.

[11] Kanjilal, P.P., and G. Saha (1995): 'Fetal ECG extraction from single channel maternal ECG using SVD and SVR spectrum', Proc. of IEEE EMBS Conference, Montreal, Sept.

[12] Longini, R.L., J.P. Giolmi, C. Wall III, and R.F. Quick (1975): 'Filtering without phase shift', *IEEE Trans. Biomedical Eng.*, **BME-22**, 432-433.

[13] Mendel, J.M. (1987): *Lessons in Digital Estimation Theory*, Prentice-Hall, Englewood Cliffs, N.J.

[14] Maybeck, P.S. (1982): *Stochastic Models Estimation and Control*, Vol. 2, Academic Press, New York.

[15] Meditch, J.S. (1969): *Stochastic Optional Linear Estimation and Control*, McGraw-Hill, New York.

[16] Rauch, H.E., F. Tung,and C.T. Streibel (1965): 'Maximum likelihood estimates of linear dynamic systems', *AIAAJ*, 3(8), 1445-1450.

[17] Vandershoot, J., D. Callaerts, W. Sansen, J. Vandewalle, G. Vantrappen, and J. Janssens (1987): 'Two methods for

optimal MECG elimination and FECG detection from skin electrode signals', *IEEE Trans. Biomedical Eng.*, **BME 34**, 233–243.

[18] Watanabe, K. (1986): 'A new forward pass fixed-interval smoother using the U-D information matrix factorization', *Automatica*, **22**(4), 465–475.

[19] Widrow, B., J.M. McCool, J. Kanmitz, C. Williams, R. Hearn, J. Zeidler, E. Dong, and R. Goodlin (1975): 'Adaptive noise cancelling: principles and applications', *Proc. IEEE*, **63**(12), 1692–1716.

APPENDIX 1

VECTOR AND MATRIX OPERATIONS

1A.1 Basic Definitions

An mxn matrix **A** is a rectangular array of elements a_{ij} arranged in m rows and n columns:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{mn} & a_{m2} & \cdots & a_{mn} \end{bmatrix};$$

A can also be expressed as a column vector

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{bmatrix},$$

where each element of the column will represent a row vector, for example

$$\mathbf{a}_2 = [a_{21} \;\; a_{22} \cdots a_{2n}].$$

Two matrices **A** and **B** can be added (or subtracted from each other), if they are of the same size:

$$\mathbf{A} + \mathbf{B} = \mathbf{C};$$

the resulting matrix **C** will also be of the same size as **A** and **B**.

The *transpose* of an mxn matrix **A** is given by the nxm matrix $\mathbf{A}^T = \mathbf{B}$ which satisfies the relationship:

$$a_{ij} = b_{ji}, \qquad 1 \le i \le m, \qquad 1 \le j \le n,$$

where a_{ij} and b_{ij} are the elements of **A** and **B** respectively. The following transpose relations also hold

(a) $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$,

(b) $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$,

(c) $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^T = \begin{bmatrix} \mathbf{A}^T & \mathbf{C}^T \\ \mathbf{B}^T & \mathbf{D}^T \end{bmatrix}$.

An mxn matrix **A** is *symmetric* if $\mathbf{A} = \mathbf{A}^T$, that is $a_{ij} = a_{ji}$. A matrix is said to be *skew symmetric* if $\mathbf{A} = -\mathbf{A}^T$, that is $a_{ij} = -a_{ji}$, and the diagonal elements are all equal to zero.

1A.2 Matrix Multiplication

In matrix multiplication, the basic operation is the multiplication of a row vector with a column vector of the same size, e.g.

$$[a\ b\ c]\begin{bmatrix} d \\ e \\ f \end{bmatrix} = ad + be + cf.$$

The product is a scalar, which is referred to as the *inner product* of the two vectors.

If the inner product of two vectors is zero, the vectors are *mutually orthogonal.* If an mxn matrix **A** multiplies with an nxr matrix **B** to produce the mxr matrix **C**, then the elements of **C** are given by

$$c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj},$$

where a_{ik} and b_{kj} are the elements of matrix **A** and **B** respectively.

Matrix multiplication depends on the order of the matrices. **AB** is known as premultiplication of **B** by **A** or as postmultiplication of **A** by **B**. The multiplication

$$\mathbf{C} = \mathbf{AB}$$

is possible only if the row length of **A** is the same as the column length of **B**. In general **AB** ≠ **BA**, even if both the matrices **A** and **B** are square.

Some general rules of matrix multiplication follow:

(a) **A(BC) = (AB)C** (associative property).
(b) **A(B+C) = AB + AC** (distributive property).
(c) in general, **AB ≠ BA** (usually not commutative).
(d) **AB = O**, does not imply **A** or **B** to be equal to **O**.

If a column vector of length m is multiplied with a row vector of length n, the product is a matrix of size mxn:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} [d\ e\ f\ g] = \begin{bmatrix} ad & ae & af & ag \\ bd & be & bf & bg \\ cd & ce & cf & cg \end{bmatrix}.$$

This result is referred to as an *outer product* or a *dyadic product.*

1A.3 Determinant

Every square matrix has a determinant, which is a scalar

quantity computed as follows.
(a) For a 2x2 matrix **A**:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

the determinant, indicated as $|\mathbf{A}|$, is given by

$$|\mathbf{A}| = a_{11}a_{22} - a_{21}a_{12}.$$

(b) If **A** is a 3x3 matrix:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

$$|\mathbf{A}| = a_{11}(a_{22}a_{33} - a_{32}a_{23}) + a_{12}(a_{23}a_{31} - a_{33}a_{21}) + a_{13}(a_{21}a_{32} - a_{31}a_{22}).$$

The determinant of a matrix **A** must be nonzero for it to have an inverse $\mathbf{A}^{-1}$. There are three basic properties of determinants:

(i) If **A** and **B** are both square matrices, $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$.

(ii) $|\mathbf{A}| = |\mathbf{A}^T|$,

(iii) $\left|\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ 0 & \mathbf{A}_4 \end{bmatrix}\right| = |\mathbf{A}_1||\mathbf{A}_4|$,

if $\mathbf{A}_1$ and $\mathbf{A}_4$ are square matrices.

1A.4 Matrix Inversion

If **A** is a square matrix, its inverse $\mathbf{A}^{-1}$ is defined by the property:

$$\mathbf{AA}^{-1} = \mathbf{I},\ \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}.$$

If **A** has an inverse, it is called *nonsingular*, otherwise it is called singular.

Some basic rules of matrix inversion follow.

(a) $[\mathbf{AB}]^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

(b) If $\mathbf{B} = \mathbf{A}^{-1}$, then $\mathbf{B}^T = [\mathbf{A}^T]^{-1}$.

(c) If **A** is symmetric, $\mathbf{A}^{-1}$ is symmetric.

(d) **A** is unique, if its inverse exists.

(e) $|\mathbf{AA}^{-1}| = |\mathbf{A}||\mathbf{A}^{-1}| = 1$, or $|\mathbf{A}^{-1}| = \dfrac{1}{|\mathbf{A}|}$.

Some useful matrix inverses follow.

(a) $$\begin{bmatrix} A & O \\ C & B \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & O \\ -B^{-1}CA^{-1} & B^{-1} \end{bmatrix},$$

(b) $$\begin{bmatrix} A & D \\ C & B \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}DB^{-1} \\ O & B^{-1} \end{bmatrix},$$

(c) $$\begin{bmatrix} A & D \\ C & B \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1}+ER^{-1}F & -ER^{-1} \\ -R^{-1}F & R^{-1} \end{bmatrix},$$

where $R = [B - C\,A^{-1}D]$, $E = A^{-1}D$, and $F = CA^{-1}$.

If $C = D^T$,

$$\begin{bmatrix} A & D \\ D^T & B \end{bmatrix} = \begin{bmatrix} A^{-1}+ER^{-1}F & -ER^{-1} \\ -(ER^{-1})^T & R^{-1} \end{bmatrix}.$$

Matrix inversion lemma

If **A** and **C** are nonsingular square matrices,

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B\,[C^{-1} + DA^{-1}B]^{-1}\,DA^{-1}.$$

This result can be shown to be valid by direct substitution as follows. Let

$$DA^{-1}B = M$$

$$\begin{aligned}
&[A + BCD]\,[A^{-1} - A^{-1}B\,[C^{-1}+ DA^{-1}B]^{-1}DA^{-1}] \\
&\quad = I + BCDA^{-1} - B[C^{-1}+ M]^{-1}DA^{-1} - BCM[C^{-1}+ M^{-1}]^{-1}DA^{-1} \\
&\quad = I + BCDA^{-1} - BC[C^{-1}+ M][C^{-1}+ M^{-1}]^{-1}DA^{-1} \\
&\quad = I + BCDA^{-1} - BCDA^{-1} \\
&\quad = I.
\end{aligned}$$

Corrollaries

(a) If **A** and **C** are nonsingular square matrices,

$$[A - BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B - C^{-1}]^{-1}DA^{-1}.$$

(b) $[I + PBCD]^{-1}P = P - PB[DPB + C^{-1}]^{-1}DP,$

where **C** is a nonsingular square matrix and **P** is a square positive semidefinite matrix.

(c) $$[A + uv^T]^{-1} = A^{-1} - \frac{(A^{-1}u)\;(V^TA^{-1})}{1 + V^TA^{-1}u},$$

where **A** is a nonsingular square matrix and **u** and **v** are column vectors.

1A.5 Some Special Forms

A matrix **U** is called *upper triangular*, if all the entries below the principal diagonal are zero:

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix};$$

U need not be a square matrix. Similarly when all elements above the principal diagonal are zeros, the matrix is called *lower triangular*.

A matrix **D** is called *diagonal*, if all entires other than the diagonal ones are zero:

$$\mathbf{D} = \begin{bmatrix} d_{11} & 0 & 0 & 0 \\ 0 & d_{22} & 0 & 0 \\ 0 & 0 & d_{33} & 0 \end{bmatrix}.$$

The inverse of a diagonal matrix is a matrix with reciprocal diagonal terms:

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{d_{11}} & 0 & 0 \\ 0 & \frac{1}{d_{22}} & 0 \\ 0 & 0 & \frac{1}{d_{33}} \\ 0 & 0 & 0 \end{bmatrix}.$$

If all the diagonal entries of a square matrix are unity, the matrix is called an *identity matrix*:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

If the rows (or columns) of an identity matrix are interchanged, it becomes a *permutation matrix*. When a matrix **A** is postmultiplied by a permutation matrix **P**, the columns of **A** are interchanged in the same order as the identity matrix rows are interchanged into **P**. For example, let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix};$$

so

$$\mathbf{AP} = \begin{bmatrix} a_{13} & a_{11} & a_{12} \\ a_{23} & a_{21} & a_{22} \\ a_{33} & a_{31} & a_{32} \end{bmatrix}.$$

The interchanging will be row-wise if **A** is premultiplied by **P**.

1A.6 Norms

The magnitude of a vector is expressed in terms of its norm. For any real vector **x**,

$$\mathbf{x} = [x_1 \; x_2 \; \dots \; x_m],$$

its length or norm, denoted by $\|\mathbf{x}\|$, is defined as

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}} = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}.$$

The vector norm is also called the *Euclidean norm*. The vector norm has the following properties.

(a) $\|\alpha\mathbf{x}\| = \|\alpha\|.\|\mathbf{x}\|$ for all real α and all vectors **x**.

(b) $\|\mathbf{x}\| > 0$ unless $\mathbf{x} = \mathbf{0}$ when $\|\mathbf{x}\| = 0$.

(c) $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all vectors **x**, **y**.

(d) $\|\mathbf{x}\| = \|\mathbf{U}\mathbf{x}\|$, for all orthogonal matrices **U**, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.

The norm and the inner product are related through the *Schwarz inequality* as

$$\|\mathbf{x}^T\mathbf{y}\| \leq \|\mathbf{x}\|\cdot\|\mathbf{y}\|,$$

for any vectors **x** and **y**.

Similar to the vector norm, matrix norms provide a measure of the magnitude of the matrix. The norm of an nxn real matrix **A** is defined as

$$\|\mathbf{A}\| = \max \|\mathbf{A}\mathbf{x}\|,$$

for all vectors **x**. The matrix norm has the following properties

(a) $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|.\|\mathbf{x}\|$,

(b) $\|\mathbf{AB}\| \leq \|\mathbf{A}\|.\|\mathbf{B}\|$,

(c) $\|\mathbf{A}+\mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$,

for all matrices **A** and **B** and vector **x**.

There are many other norms that can be defined for vectors and matrices. Some of the popularly used norms are as follows.

(a) *Frobenius norm* of any mxn matrix **A** is expressed as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}|a_{ij}|^2}$$

(b) *1-norm* and *∞-norm* of n-vector **x** are given by

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n}|x_i|,$$

$$\|\mathbf{x}\|_\infty = \max_{1\le i\le n}|x_i|,$$

1-norm and ∞-norm of mxn matrix **A** are given by

$$\|\mathbf{A}\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}|a_{ij}|,$$

$$\|\mathbf{A}\|_\infty = \max_{1\le i\le m}\sum_{j=1}^{n}|a_{ij}|.$$

(c) *2-norm* of an mxn matrix **A** is given by the square root of the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$, that is the largest singular value of **A**.

1A.7 Differentiation

If mxn matrix **A** (=((a_{ij}))) is a function of a variable t,

$$\frac{d\mathbf{A}}{dt} = \begin{bmatrix} \frac{da_{11}}{dt} & \frac{da_{12}}{dt} & \cdots & \frac{da_{1n}}{dt} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{da_{1m}}{dt} & \frac{da_{2m}}{dt} & \cdots & \frac{da_{mn}}{dt} \end{bmatrix}.$$

If m-column vector **x** is a function of n-row vector **y**, where

$$\mathbf{x} = [x_1 \;\; x_2 \cdots x_m]^T,$$

$$\mathbf{y} = [y_1 \;\; y_2 \cdots y_m],$$

then

$$\frac{\partial \mathbf{x}}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial y_1} & \frac{\partial \mathbf{x}}{\partial y_2} & \cdots & \frac{\partial \mathbf{x}}{\partial y_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \cdots & \frac{\partial x_1}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial y_1} & \frac{\partial x_m}{\partial y_2} & \cdots & \frac{\partial x_m}{\partial y_n} \end{bmatrix}.$$

The matrix or vector integration can be similarly expressed.

1A.8 Matrix operations

Matrix operations appear widely in the present text explicitly or implicitly. Some of the prime operations are listed in the following table

Table 1A.8 Some common matrix operations

Matrix operations	Expression	Typical applications
Cholesky factorization of A^TA, A being mxn matrix	$A^TA = LDL^T = GG^T$ L is lower triangular D is diagonal G is lower triangular	Matrix inversion, LS estimation
Singular value decomposition of mxn matrix A	$A=USV^T$, U and V are orthogonal matrices, S is diagonal	Matrix rank determination, invertion,data compression signal decomposition, modelling, filtering estimation and prediction
QR decomposition on mxn matrix A	A = QR, Q having orthogonal columns, R is upper triangular	Matrix inversion, LS estimation
QRcp factorization of A	$Q^T[A]P = R$ Q having orthonormal columns, R is upper triangular, P is permutation matrix	Subset selection (first r columns of AP)
UD factorization of square symmetric matrix P	$P = UDU^T$. U is upper triangular D is diagonal	Recursive updation of covariance matrix, Inversion of functions of matrices

Bibliography

[1] Bellman, R.E. (1960): *Introduction to Matrix Theory*, McGraw-Hill, New York.

[2] Forsythe, G.E., and C.B.Moler (1967): *Computer Solution to Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

[3] Golub, G.H., and C.F. Van Loan (1989): *Matrix computations*, Johns Hopkins Univ. Press, Baltimore.

[4] Nobel, B., and J.W. Daniel (1988): *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, New Jersey.

[5] Rice, J.R. (1981): *Matrix Computations and Mathematical Software*, McGraw-Hill, New York.

[6] Strang, G. (1988): *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, San Diego.

APPENDIX 2

EXPONENTIAL FOURIER SERIES

The trigonometric Fourier series representation of a periodic process f(t), with period T, was discussed in Sec.2.5.1. The derivation of exponential Fourier series representation is presented here.

The sinusoidal functions can be expressed in terms of exponential functions as follows

$$e^{i\theta} = \cos\theta + i\sin\theta,$$

$$\cos\theta = \frac{1}{2}(e^{i\theta} + e^{-i\theta}), \qquad \sin\theta = \frac{1}{2}(e^{i\theta} - e^{-i\theta}), \qquad i = \sqrt{-1}.$$

So the trigonometric Fourier series (2.5.2) can be expressed as

$$f(t) = a_o + \frac{1}{2}\sum_{n=1}^{\infty}\{a_n(e^{in\omega_o t}+e^{-in\omega_o t}) - ib_n(e^{in\omega_o t}-e^{-in\omega_o t})\}, \tag{2A.1}$$

where $\omega_o=2\pi/T$. Introduce the complex coefficients:

$$g_n = \frac{1}{2}(a_n-ib_n), \quad g_{-n} = \frac{1}{2}(a_n+ib_n), \tag{2A.2}$$

Substituting for a_n and b_n from (2.5.4) in (2A.2)

$$g_n = \frac{1}{T}\int_{t_o}^{t_o+T} f(t)(\cos n\omega_o t - i\sin n\omega_o t)dt$$

$$= \frac{1}{T}\int_{t_o}^{t_o+T} f(t)e^{-in\omega_o t}dt.$$

where t_o is arbitrary. For n = 0,

$$g_o = \frac{1}{T}\int_{t_o}^{t_o+T} f(t)dt = a_o,$$

as in (2.5.4). Hence, from (2A.2) and (2A.1),

$$f(t) = a_o + \sum_{n=1}^{\infty}(g_n e^{in\omega_o t} + g_{-n}e^{-in\omega_o t}).$$

That is

$$f(t) = \sum_{n=-\infty}^{\infty} g_n e^{in\omega_o t}. \tag{2A.3}$$

Note that the magnitude and the phase of the components of the exponential series (2A.3) are given by

$$g_n = |g_n| e^{i\phi_n}, \qquad \text{and} \qquad g_{-n} = |g_n| e^{-i\phi_n}.$$

Hence

$$f(t) = g_o + \sum_{n=1}^{\infty} \{ |g_n| \, e^{\,i(n\omega_o t + \phi_n)} + |g_n| e^{-i(n\omega_o t + \phi_n)} \}.$$

That is

$$f(t) = g_o + \sum_{n=1}^{\infty} 2|g_n| \cos(n\omega_o t + \phi_n). \tag{2A.4}$$

In comparison with (2.5.5), the magnitude $|g_n|$ of the frequency component $n\omega_o$ in the exponential series (2A.4) is half of that of the sinusoidal component for the same frequency, except for the average or constant component which is the same in both.

The total power in f(t) within the interval (t_o, t_o+T), is given by the Parseval's Theorem (Papoulis, 1991) as

$$\int_{t_o}^{t_o+T} f^2(t)dt = a_o^2 + \frac{1}{2}\sum_{n=1}^{\infty} (a_n^2 + b_n^2) \tag{2A.5}$$

$$= g_o^2 + 2\sum_{n=1}^{\infty} |g_n|^2.$$

Reference

[1] Papoulis, A. (1991): *Probability, Random Variables, and Stochastic Processes*, 3rd edn., Mcgraw-Hill, New York.

Appendix 3A

U-D COVARIANCE MEASUREMENT UPDATE

3A.1 Covariance Measurement Update

In sequential state estimation, the covariance of the estimation error undergoes two different updates: (i) the measurement update, and (ii) the time update. The generic expressions for these two updates with respect to the Kalman filter state estimator are discussed in Sec.6.6. These updates appear in different areas of estimation and control.

In RLS parameter estimation, at each time instant k, the covariance **P**(k) is to be updated. The covariance measurement update produces the Kalman gain **k**(k) based on **P**(k-1), and then updates **P**(k-1) to **P**(k). RLS estimator uses **k**(k) to update the parameter estimates θ(k) as:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{k}(k)(y(k) - \mathbf{h}^T(k)\hat{\theta}(k-1)), \qquad (3A.1)$$

where y(k) is the newly available measurement and **h**(k) is the data vector.

Given the square symmetric covariance matrix **P**(k-1) and **h**(k), the covariance measurement update computes Kalman gain **k**(k) and the updated covariance matrix **P**(k):

$$\mathbf{k}(k) = \mathbf{P}(k-1)\mathbf{h}(k)(\lambda+\mathbf{h}^T(k)\mathbf{P}(k-1)\mathbf{h}(k))^{-1}, \qquad (3A.2)$$

and

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{k}(k)\mathbf{h}^T(k)]\mathbf{P}(k-1). \qquad (3A.3)$$

Refer to Sec.3.4.1 for further discussions. One of the popular approaches for covariance measurement update is the U-D factorization; detailed discussion of U-D covariance factorization is given in Bierman (1977), and Thornton and Bierman (1980).

Remarks

(a) With reference to the Kalman filter formulation (6.6.9-6.6.13), the RLS parameter estimation is equivalent to the state estimation problem for the process

$$\mathbf{x}(k+1) = \mathbf{x}(k), \qquad (3A.4)$$

$$y(k) = \mathbf{h}^T(k)\mathbf{x}(k) + e(k), \qquad (3A.5)$$

where **x** is the same as the parameter vector θ in (3A.1). The relation (3A.4) renders the covariance time update

redundant, and hence RLS estimation requires the covariance measurement update only.
(b) Here **k**(k) is in fact **k**(k|k-1), as all information up to time (k-1) is used in producing **k**(k).

3A.2 U-D Factorization

The positive semidefinite matrix **P** can be expressed as

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T = [\mathbf{U}\mathbf{D}^{1/2}][\mathbf{U}\mathbf{D}^{1/2}]^T, \tag{3A.6}$$

where **D** is a diagonal matrix, and **U** is an upper triangular matrix with 1 s on the diagonal; $\mathbf{U}\mathbf{D}^{1/2}$ is the square root of **P**. The factorization (3A.6) is referred to as U-D factorization of the covariance matrix. So instead of recursively updating **P**, its factors **U** and **D** may be updated and propagated through the recursions.

U-D factorization belongs to the class of square-root filtering, where instead of the covariance matrix **P**, its square-root factors are updated. This approach reduces round-off errors and increases numerical stability. U-D factorization offers the special advantages that
(a) no explicit square-root extractions are required, and
(b) **D** is available for diagnostic purposes.

3A.3 Fortran mechanization of U-D measurement update

Bierman (1977, p.100) considers **U** stored as a matrix; the present implementation considers the relevant upper triangular part of **U** stored into a vector U(), typically as

$$\mathbf{U} = \begin{bmatrix} 1 & U(1) & U(2) & U(4) \\ & 1 & U(3) & U(5) \\ & & 1 & U(6) \\ 0 & & & 1 \end{bmatrix}.$$

The present routine produces *unweighted* Kalman gain $\mathbf{k}'(k) = \mathbf{P}(k-1)\mathbf{h}(k)$, and ALPHAJ = $(1+\mathbf{h}^T(k)\mathbf{P}(k-1)\mathbf{h}(k))$ which is a scalar; the actual Kalman gain $\mathbf{k}(k) = \mathbf{k}'(k)/\text{ALPHAJ}$.

Inputs:	X()	data vector **h**; remains undestroyed on exit
	U()	Prior U factor arranged into a vector
	D()	Prior D factor, expressed as a vector
	NPAR	Size of X()
	FLAMDA	scalar cost λ; FLAMDA=1 for estimation
	y	measurement of the dependent variable
	THETA()	Parameter vector θ

Outputs: K() Unweighted Kalman gain
ALPHAJ Scalar factor; Kalman gain = K()/ALPHAJ
U() Updated U factor
D() Updated D factor.
THETA() Updated parameter vector

Measurement update routine

```
      FJ = X(1)
      VJ = D(1)*FJ
      K(1) = VJ
      ALPHAJ = FLAMDA + VJ*FJ
```

Comment: If FLAMDA is zero and if VJ or FJ are also zero, ALPHAJ becomes zero, which causes numerical problems at the next step. So, if FJ or VJ are likely to be zero, the lowest acceptable value for FLAMDA may be constrained to a very small number, say close to the machine zero, instead of true zero (e.g., IF(FLAMDA.LT.1E-14) FLAMDA=1E-14).

```
      D(1) = D(1)*FLAMDA/ALPHAJ
      KF = 0
      KU = 0
      DO 4 J = 2,NPAR
      FJ = X(J)
      DO 41 I = 1,J-1
      KF = KF+1
41    FJ = FJ + X(I)*U(KF)
      VJ = FJ * D(J)
      K(J) = VJ
      AJLAST = ALPHAJ
      ALPHAJ = AJLAST + VJ*FJ
      D(J) = (D(J)*AJLAST)/(ALPHAJ*FLAMDA)
      PJ  = -FJ/AJLAST
      DO 42 I = 1,J-1
      KU = KU+1
      W = U(KU) + K(I)*PJ
      K(I) = K(I) + U(KU)*VJ
42    U(KU) = W
4     CONTINUE
```

Comment: The following part is required in the case of RLS parameter estimation.

```
      PERR = y
      DO 6 I = 1, NPAR
```

```
6     PERR = PERR - THETA(I)*X(I)
      DO 8 I = 1,NPAR
8     THETA(I) = THETA(I) + PERR*K(I)/ALPHAJ
```

3A.4 Application Aspects

Besides RLS parameter estimation, the use of U-D measurement update in LQ control (Sec.11.6.3, Equation (11.6.37), and Sec.11.6.4), in generalized predictive control (Sec.12.8, Eqn.(12.8.3 - 12.8.4)), and in Koopmans-Levin method of parameter estimation (Sec.3.5.3, Eqn.(3.5.21)) have already been discussed. Further applications of U-D measurement update for matrix operations are possible as shown below.

Example 3A.3 Compute $[\mathbf{G}^T\mathbf{G} + \mathbf{M}^T\mathbf{M}]^{-1}\mathbf{t}$, where **G** and **M** are nxn matrices and **t** is a vector, using U-D measurement update routine.

Following (3.4.8 - 3.4.9), RLS estimation involves the update

$$\mathbf{P}(k+1)^{-1} = [\mathbf{P}(k)]^{-1} + \mathbf{h}(k)\mathbf{h}^T(k).$$

So, $[\mathbf{G}^T\mathbf{G} + \mathbf{M}^T\mathbf{M}]^{-1}$ can be computed as follows. Initialize $\mathbf{P} = \lambda\mathbf{I}$, where λ is vanishingly small (say, 1E-14); fill the U() array with all zeros and D() array with λ values. If $\mathbf{M} = [\mathbf{m}_1 \ldots \mathbf{m}_i \ldots \mathbf{m}_n]^T$, n calls to U-D measurement update, with $\mathbf{m}_i$ (i=1 to n) passed as the data vector X(), will produce $\mathbf{P} = \mathbf{UDU}^T = [\mathbf{M}^T\mathbf{M}]^{-1}$. Similarly further n calls to the U-D routine with each column of $\mathbf{G}$ passed as the data vector X(), will lead to the update $\mathbf{P} = [\mathbf{G}^T\mathbf{G} + \mathbf{M}^T\mathbf{M}]^{-1}$.

Again the present routine produces

$$\mathbf{k}'(k) = \mathbf{P}(k-1)\mathbf{h}(k),$$

which can be used next as follows. Define X() as the vector t, and run the UD-measurement update once; K() will contain the result $[\mathbf{G}^T\mathbf{G} + \mathbf{M}^T\mathbf{M}]^{-1}\mathbf{t}$.

References

[1] Bierman, G.J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[2] Thornton, C.L., and G.J. Bierman (1980): 'UDU^T covariance factorization for Kalman filtering', *Control and Dynamic Systems - Advances in theory and applications*, C.T. Leondes (Ed.), 16, 178-248.

Appendix 3B

QR AND QRcp FACTORIZATION

The **QR** decomposition of an mxn matrix **A** with rank p is given by

$$\mathbf{A} = \mathbf{QR}$$

where **Q** is an mxp matrix with orthonormal columns and **R** is an pxn upper triangular matrix. When m = n, **Q** and **R** are square matrices, and **Q** is an orthogonal matrix. There are three main approaches to QR decomposition: Gram-Schmidt orthogonalization, Householder transformation, and Givens rotation. The Gram-Schmidt orthogonalization has been used in the present discussion. This appendix also presents the mechanism for QR with column pivoting (QRcp) factorization.

QR factorization through Gram-Schmidt orthogonalization

Starting with mxm matrix **A**, the Gram-Schmidt orthogonalization process produces matrix $\mathbf{Q} = [\mathbf{q}_1 \ \mathbf{q}_2 \cdots \mathbf{q}_m]^T$ having orthonormal column vectors, $\mathbf{q}_i$ (that is $\mathbf{q}_i^T\mathbf{q}_i = 1$, for i = 1 to m).

Consider a 3x3 matrix **A**:

$$\mathbf{A} = \begin{bmatrix} | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\ | & | & | \end{bmatrix},$$

from which

$$\mathbf{Q} = \begin{bmatrix} | & | & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \\ | & | & | \end{bmatrix},$$

with $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, is to be produced.

The first column vector $\mathbf{q}_1$ is considered to be spanning the same space as $\mathbf{a}_1$; hence

$$\mathbf{q}_1 = \mathbf{a}_1/\|\mathbf{a}_1\|. \qquad (3B.1)$$

To determine second vector $\mathbf{q}_2$, start with $\mathbf{a}_2$ and deduct the component of $\mathbf{a}_2$ in the direction of the first vector (i.e., $\mathbf{a}_1$ or $\mathbf{q}_1$) as follows:

$$\mathbf{q}_2' = \mathbf{a}_2 - (\mathbf{q}_1^T\mathbf{a}_2)\mathbf{q}_1;$$

on subsequent normalization

$$q_2 = \frac{q_2'}{\|q_2'\|}. \tag{3B.2}$$

To compute q_3, which will be orthogonal to both q_1 and q_2, the components of a_3 in the spaces spanned by q_1 and q_2 have to be subtracted from a_3 as follows:

$$q_3' = a_3 - (q_1^T a_3)q_1 - (q_2^T a_3)q_2;$$

on subsequent normalization

$$q_3 = \frac{q_3'}{\|q_3'\|}. \tag{3B.3}$$

Thus the Gram-Schmidt orthogonalization of **A** produces matrix **Q** with orthogonal columns. The **QR** decomposition follows naturally from the Gram-Schmidt orthogonalization as follows.

Denote

$$r_{11} = \|a_1\|,$$

$$r_{12} = q_1^T a_2, \quad r_{22} = \|q_2'\|,$$

$$r_{13} = q_1^T a_3, \quad r_{23} = q_2^T a_3, \qquad r_{33} = \|q_3'\|.$$

So the relationship between the column vectors of **A** and **Q** can be rewritten as

$$q_1 = \frac{a_1}{r_{11}}, \tag{3B.4}$$

$$q_2 = \frac{a_2}{r_{22}} - \frac{r_{12}q_1}{r_{22}}, \tag{3B.5}$$

$$q_3 = \frac{a_3}{r_{33}} - \frac{r_{13}q_1}{r_{33}} - \frac{r_{23}q_2}{r_{33}}. \tag{3B.6}$$

Again the vectors a_i can be expressed in terms of q_i as follows.

$$a_1 = r_{11}\, q_1,$$

$$a_2 = r_{12}q_1 + r_{22}q_2,$$

$$a_3 = r_{13}q_1 + r_{23}q_2 + r_{33}q_3,$$

or

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ 0 & & r_{33} \end{bmatrix}.$$

Generalizing (3B.4 - 3B.6)

$$q_j = \frac{a_j}{r_{jj}} - \frac{r_{1j}q_1}{r_{jj}} - \frac{r_{2j}q_2}{r_{jj}} - \ldots - \frac{r_{(j-1)j}q_{j-1}}{r_{jj}},$$

where

$$r_{ij} = q_i^T a_j \quad \text{and}$$

$$r_{jj} = \|(a_j - r_{1j}q_1 - r_{2j}q_2 - \cdots - r_{(j-1)}q_{j-1})\|.$$

Remarks

(1) For any mxn matrix **A** = $[a_1, \ldots, a_n]$, the corresponding **Q** with the orthonormal vectors **Q** = $[q_1, \ldots, q_n]$ are such that for r = 1, ..., n, the set $\langle q_1, \ldots, q_r \rangle$ collectively span the same r dimensional space as the set $\langle a_1, \ldots, a_r \rangle$.

(2) The number of nonzero diagonal elements of **R** indicates the rank of the matrix, although the closeness to rank-loss cannot be detected through QR factorization as precisely as through SVD. **R**(j,j) being zero indicates the j-th column of **A** being redundant, as it has no component in the q_j vector space which is orthogonal to the q_i vector spaces for i ≠ j.

(3) Although the mechanism of orthogonalization is quite transparent in Gram-Schmidt orthogonalization, from a numerical point of view it is not well conditioned. A numerically sound implementation is *modified weighted* Gram-Schmidt algorithm; this as well as other robust implementations are detailed in Golub and Van Loan (1989) and Lawson and Hanson (1974).

QRcp Factorization

QRcp (that is, QR with column pivoting) factorization is used to pivote the columns of a matrix in order of maximum Euclidian norm in successive orthogonal directions, while QR factorization is performed on the matrix. The implementation of QRcp factorization has been discussed in Sec.3.6.2. The mechanism of the rotation of the columns is discussed here.

The column vector of **A** with $\max(\mathbf{a}_i^T\mathbf{a}_i)$ is first selected, and is swapped with $\mathbf{a}_1$. $\mathbf{q}_1$, the unit vector in the direction of $\mathbf{a}_1$ is determined using (3B.1).

The second (or rotated) vector is the one maximizing

$$(\mathbf{a}_j-\mathbf{q}_1^T\mathbf{a}_j\mathbf{q}_1)^T(\mathbf{a}_j-\mathbf{q}_1^T\mathbf{a}_j\mathbf{q}_1),$$

which is swapped with $\mathbf{a}_2$, and $\mathbf{q}_2$ is computed as (3B.2).

At the i-th stage of selection, the rotated vectors $(\mathbf{a}_j^*)$ are

$$\mathbf{a}_j^* = \mathbf{a}_j - (\mathbf{q}_1^T\mathbf{a}_j\mathbf{q}_1+\ldots+\mathbf{q}_{i-1}^T\mathbf{a}_j\mathbf{q}_{i-1}),$$

$$i = 2 \text{ to } n,\ j = i \text{ to } n,$$

and the i-th selected vector is the one maximizing $\mathbf{a}_j^{*T}\mathbf{a}_j^*$. The subsequent rotation within QR decomposition will be with respect to this vector and so on. The selection is continued for up to r stages, where r may be the rank of **A** or may be specified based on other considerations (see for example Sec.3.6.3). The sequence of successive selections is registered in the permutation matrix **P**. The result is

$$\mathbf{Q}^T\mathbf{AP} = \mathbf{R},$$

where **R** is upper triangular. The matrix **AP** will have the r columns of **A** appearing first in order of importance.

References

[1] Golub, G.H., and C.F. Van Loan (1989): *Matrix Computations*, 2nd edn., The Johns Hopkins Univ. Press, Baltimore

[2] Lawson, C.L., and Hanson, R.J. (1974): *Solving Least Squares Problems*, Prentice Hall, Englewood Cliffs, N.J.

APPENDIX 4

CENTRED MOVING AVERAGE

The two basic purposes of Centred Moving Averaging (CMA) are

(i) estimation of trend in a time series without any time-lag, and

(ii) reduction of the effects of random or spurious noise associated with the data.

From the signal processing point of view, CMA is akin to smoothing or estimation of $\hat{y}(k)$ based on the data $\langle y(k-j), \ldots, y(k+i)\rangle$, i and j being positive integers.

The general consequence of *averaging* is low-pass filtering (see Appendix 14A), that is the high frequency components are attenuated whereas the low frequency components are retained. Any averaging which uses present and the past data only will produce a time-lag in the averaged data. In CMA, since both past and post data are used, the lag-free estimate of the series is produced. Thus CMA is similar to bidirectional filtering discussed in Sec.14.3.1.

In the usual form of CMA, the estimate of the time series $\langle y(.)\rangle$ at time k is given by

$$\hat{y}(k) = \frac{y(k-N)+\ldots+y(k-1)+y(k)+y(k+1)+\ldots+y(k+N)}{2N+1}; \qquad (4A.1)$$

here (2N+1) is the length of the data window, which characterizes the CMA.

The data may be exponentially weighted as follows:

$$\hat{y}(k) = \frac{\alpha^N y(k-N)+\ldots+\alpha y(k-1)+y(k)+\alpha y(k+1)+\ldots+\alpha^N y(k+N)}{(2N+1)(1+2(\alpha+\alpha^2+\ldots+\alpha^N))},$$

$$0<\alpha\leq 1.$$

The implication is that more importance is given to the value of the data closer to k, the point of estimation.

If the data series has an inherent periodic component of length N, CMA through (4A.1) will be able to get rid of the periodicity, when N is odd. If N is even, use the scheme

$$\hat{y}(k)=\frac{1}{2}\left[\left(\frac{y(k-N/2)+..+y(k+N/2-1)}{N}\right)+\left(\frac{y(k-N/2-1)+..+y(k+N/2)}{N}\right)\right]. \qquad (4A.2)$$

From the low-pass filtering point of view, CMA with an averaging window of N will eliminate any periodic component with period length N, or its higher harmonics of period

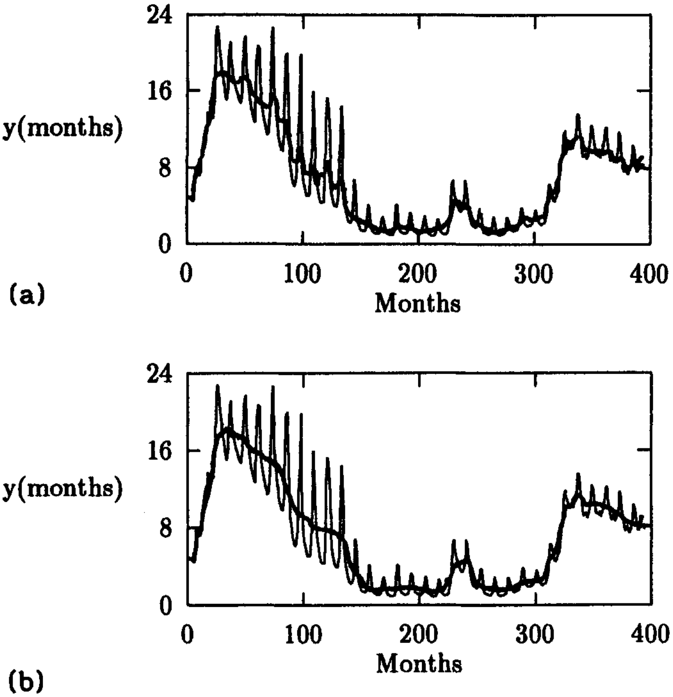


Figure 4A.1 Central moving averaging of the German unemployment series with (a) N = 10, (b) N = 12. ——— actual data, ——— filtered data $\{\hat{y}(k)\}$.

length N/2, N/3, etc. For an even N, the scheme as in (4A.2) ensures $\hat{y}(k)$ to be truly centred over the time-point k.

CMA is discussed in detail in Makridakis *et al* (1983).

Example 4A.1 Centred moving averaging of the German unemployment series (Appendix 7E)

This monthly data series (see Fig.14.3.3) has a yearly periodicity, i.e. the period length concerned is 12. CMA using (4A.2) with N=10 and N=12 is shown in Fig.4A.1. As expected, unlike the case with N=12, CMA with N=10 fails to eliminate the periodic components completely.

Reference

[1] Makridakis, S., S.C. Wheelwright, and V.E. McGee (1983): *Forecasting: Methods and Applications*, 2nd. edn., John Wiley, New York.

APPENDIX 5A

RECURSION OF THE DIOPHANTINE EQUATION

5A.1 Problem Statement

Discrete-time modelling and prediction using transfer-function models often involve solution of the Diophantine equation:

$$C(q^{-1}) = E_p(q^{-1})A(q^{-1}) + q^{-p}F_p(q^{-1}), \qquad p = 1,2,\ldots,$$

where

$$\begin{aligned}
A(q^{-1}) &= 1 + a_1q^{-1} + \ldots + a_nq^{-n}, \\
C(q^{-1}) &= 1 + c_1q^{-1} + \ldots + c_nq^{-n}, \\
E_p(q^{-1}) &= 1 + e_{(p)1}q^{-1} +\ldots+ e_{(p)p-1}q^{-p+1}, \\
F_p(q^{-1}) &= f_{(p)0} + f_{(p)1}q^{-1} +\ldots+ f_{p(n-1)}q^{-n+1}.
\end{aligned}$$

it is assumed that $A(q^{-1})$ and $C(q^{-1})$ are known; the objective is to determine $E_p(q^{-1})$ and $F_p(q^{-1})$ for different values of p recursively.

5A.2 Recursive Solution

For p = 1,

$E_p(q^{-1}) = 1$ and hence

$$q^{-1}F_p(q^{-1}) = C(q^{-1}) - A(q^{-1}).$$

So, for i = 1 to n,

$$f_{(1)i-1} = c_i - a_i. \tag{5A.1}$$

For p = 2,

$E_p(q^{-1}) = 1 + e_{(2)1}q^{-1}$

$$C(q^{-1}) = A(q^{-1}) + q^{-1}\,e_{(2)1}A(q^{-1}) + q^{-2}F_p(q^{-1})$$

That is

$$\begin{aligned}
C(q^{-1}) - A(q^{-1}) &= e_{(2)1}q^{-1} + e_{(2)1}a_1q^{-2} + e_{(2)1}a_2q^{-3} + \\
&\quad \ldots + f_{(2)0}q^{-2} + f_{(2)1}q^{-3} + \ldots
\end{aligned}$$

So

$$\begin{array}{llll}
e_{(2)1} = c_1 - a_1 & & = f_{(1)0}, & \\
f_{(2)0} = c_2 - a_2 - e_{(2)1}a_1 & & = f_{(1)1} - e_{(2)1}a_1, & \\
f_{(2)1} = c_3 - a_3 - e_{(2)1}a_2 & & = f_{(1)2} - e_{(2)1}a_2. & \\
\vdots & & & \\
f_{(2)n-2} = c_n - a_n - e_{(2)1}a_{n-1} & & = f_{(1)n-1} - e_{(2)1}a_n & \\
f_{(2)n-1} = \quad - e_{(2)1}a_n & & = \quad - e_{(2)1}a_n &
\end{array} \tag{5A.2}$$

Thus the parameters of $E_p(q^{-1})$ and $F_p(q^{-1})$ can be computed recursively.

The general expression for the recursion can be developed as follows. Given the parameters of the identity

$$C(q^{-1}) = E_p(q^{-1})A(q^{-1}) + q^{-p}F_p(q^{-1}), \tag{5A.3}$$

the objective is to compute the parameters of $E_{p+1}(q^{-1})$ and $F_{p+1}(q^{-1})$ in

$$C(q^{-1}) = E_{p+1}(q^{-1})A(q^{-1}) + q^{-p-1}F_{p+1}(q^{-1}). \tag{5A.4}$$

Let $E_p(q^{-1})$, $F_p(q^{-1})$, $E_{p+1}(q^{-1})$ and $F_{p+1}(q^{-1})$ be represented by E, F, $\hat{E}$ and $\hat{F}$ respectively for simplicity.

Subtracting (5A.3) from (5A.4),

$$0 = (\hat{E}-E)A + q^{-p}(q^{-1}\hat{F}-F). \tag{5A.5}$$

$\hat{E}$ being of degree p,

$$\hat{E} - E = \bar{E} + q^{-p}\hat{e}_p, \tag{5A.6}$$

where the degree of $\bar{E}$, $\delta\bar{E} = p-1$.

From (5A.5) and (5A.6),

$$0 = \bar{E}A + q^{-p}(\hat{e}_pA + q^{-1}\hat{F} - F). \tag{5A.7}$$

Since $\bar{E}$ is of degree p-1 and the 1st term in A is 1, $\bar{E} = 0$. Hence from (5A.7),

$$q^{-1}\hat{F} = F - \hat{e}_pA,$$

or

$$\hat{e}_p = f_o, \tag{5A.8}$$

$$\hat{f}_{i-1} = f_i - \hat{e}_pa_i, \qquad i = 1,\ldots, \delta F, \tag{5A.9}$$

$$\hat{f}_i = \hat{e}_pa_n. \tag{5A.10}$$

From (5A.6), also note that

$$E_{p+1}(q^{-1}) = E_p(q^{-1}) + q^{-p}e_{(p+1)p}, \tag{5A.11}$$

that is the only additional term in $E_{p+1}(q^{-1})$ at every recursion is $\hat{e}_p$ (i.e. $e_{(p+1)p}$), given by (5A.8).

Thus, starting with (5A.1), for p = 1, the parameters of $E_{p+1}(q^{-1})$ and $F_{p+1}(q^{-1})$ can be recursively computed for higher values of p using (5A.8) – (5A.10). The recursive relationships for p = 2 are shown in (5A.2).

5A.3 Implementation

A recursive implementation of Diophantine equation in Matlab [1] follows.

```
function [E, F] = diophant(C, A, p)
LC = length(C);
LA = length(A);
if LC == LA
      [E, F] = deconv([C  zeros(1 : p - 1)], A);
elseif LC < LA
      [E, F] = deconv([C  zeros(1 : LA - LC + p - 1)], A);
elseif LC - LA - p + 1 > 0
      [E, F] = deconv([C, [A  zeros(1 : LC - LA - p + 1)]]);
else
      [E, F] = deconv([C  zeros(1 : p - 1 - LC + LA)], A);
end
LF = length(F);
F = F(p + 1 : LF);
      Lmin = p;
if LA == 1
      Lmin = min(p, LC);
end
E = E(1 : Lmin);
```

Example: see Example 12.6 in Sec.12.6.

Reference

[1] Matlab matrix software, The MathWorks, Inc., Sherborn, MA.

APPENDIX 5B

PREDICTOR FOR A MULTIVARIABLE PROCESS

A predictor algorithm formulated for a multivariable stochastic process is presented here.

Problem formulation

Consider the multi-input multi-output process:

$$\mathbf{A}(q^{-1})\mathbf{y}(k) = \mathbf{B}(q^{-1})\mathbf{u}(k-d) + \mathbf{C}(q^{-1})\mathbf{e}(k), \tag{5B.1}$$

where **y** is rxl output vector, **u** is sxl input vector; $\{\mathbf{e}(k)\}$ is a sequence of independent and equally distributed rxl random vectors with zero mean and the covariance given by

$$\mathbf{E}\{\mathbf{e}(k)\mathbf{e}^T(k)\} = \Omega.$$

A, **B** and **C** are matrix polynomials:

$$\begin{aligned}\mathbf{A}(q^{-1}) &= \mathbf{I} + \mathbf{A}_1 q^{-1} + \ldots + \mathbf{A}_n q^{-n},\\ \mathbf{B}(q^{-1}) &= \mathbf{B}_o + \mathbf{B}_1 q^{-1} + \ldots + \mathbf{B}_n q^{-n},\\ \mathbf{C}(q^{-1}) &= \mathbf{I} + \mathbf{C}_1 q^{-1} + \ldots + \mathbf{C}_n q^{-n},\end{aligned} \tag{5B.2}$$

where $\mathbf{A}_i$ and $\mathbf{C}_i$ are rxr matrices and $\mathbf{B}_i$ are rxs matrices for i = 1 to n. It is assumed that the zeros of det.$\mathbf{A}(q^{-1})$ and det.$\mathbf{C}(q^{-1})$ are strictly inside the unit circle. The input vector **u**(k-d) is defined as,

$$\mathbf{u}(k-d) = \begin{bmatrix} q^{-d_1} & & & 0 \\ & q^{-d_2} & & \\ & & \ddots & \\ 0 & & & q^{-d_s} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_s(t) \end{bmatrix}. \tag{5B.3}$$

Define $\hat{\mathbf{y}}(k+p|k)$ as the p-step ahead predictor of **y**(k), based on all the available information up to time k. If $p > d_i$ for $1 \le i \le s$, as in (5B.3), the future input strategy is assumed to be known.

The objective is to compute the minimum mean square error predictor $\hat{\mathbf{y}}(k+p|k)$ minimizing the scalar cost

$$J_m = E\{\underline{\varepsilon}^T(k+p)\underline{\varepsilon}(k+p)\}, \tag{5B.4}$$

where the prediction error

$$\underline{\varepsilon}(k+p) = \mathbf{y}(k+p) - \hat{\mathbf{y}}(k+p|k). \tag{5B.5}$$

Explicit design

Introduce the identity

$$A^{-1}(q^{-1})C(q^{-1}) = \tilde{C}(q^{-1})\tilde{A}^{-1}(q^{-1})$$
$$= E(q^{-1}) + q^{-p}F(q^{-1})\tilde{A}^{-1}(q^{-1}), \qquad (5B.6)$$

where

$$E(q^{-1}) = I + E_1q^{-1}+\ldots+ E_{p-1}q^{-p-1}, \qquad (5B.7)$$
$$F(q^{-1}) = F_o + F_1q^{-1}+\ldots+ F_{n-1}q^{-n+1}. \qquad (5B.8)$$

Note that (a) the polynomial matrices $\tilde{C}(q^{-1})$ and $\tilde{A}(q^{-1})$ in (5B.6) are not unique, and (b) det.$\tilde{C}$ = det.C. From (5B.1),

$$y(k+p) = A^{-1}Bu(k+p-d) + A^{-1}C\ e(k+p).$$

Using (5B.6),

$$y(k+p) = A^{-1}Bu(k+p-d) + Ee(k+p) + F\tilde{A}^{-1}e(k). \qquad (5B.9)$$

Substituting for e(k) in (5B.9) from (5B.1),

$$y(k+p) = A^{-1}Bu(k+p-d) + Ee(k+p) + F\tilde{A}^{-1}[C^{-1}Ay(k)-C^{-1}Bu(k-d)],$$
$$= A^{-1}Bu(k+p-d) + Ee(k+p) + F\tilde{C}^{-1}y(t) - F\tilde{A}^{-1}C^{-1}Bu(k-d),$$
$$= F\tilde{C}^{-1}y(k) + [A^{-1} - q^{-p}F\tilde{A}^{-1}C^{-1}]Bu(k+p-d) + Ee(k+p).$$

Using (5B.6),

$$y(k+p) = F\tilde{C}^{-1}y(k) + EC^{-1}Bu(k+p-d) + Ee(k+p) \qquad (5B.10)$$
$$= \tilde{y}(k+p|k) + \underline{\varepsilon}(k+p), \qquad \text{by definition.}$$

Since Ee(k+p) is orthogonal to the other terms on the right-hand-side of (5B.10) the optimal predictor is given by

$$\hat{y}(k+p|k) = FG(q^{-1})\tilde{C}^{-1}(q^{-1})y(k)+E(q^{-1})C^{-1}(q^{-1})B(q^{-1})u(k+p-d).$$

If $C(q^{-1}) = I$, $\tilde{C}(q^{-1}) = I$ and $\tilde{A}(q^{-1}) = A(q^{-1})$ as particular solutions, and the p-step ahead predictor becomes

$$\hat{y}(k+p|k) = F(q^{-1})y(k) + E(q^{-1})B(q^{-1})u(k+p-d).$$

$\underline{\varepsilon}(k+p)$, the prediction error corresponding to the optimal predictor is given by

$$\underline{\varepsilon}(k+p) = E(q^{-1})e(k+p) = e(k+p) + E_1e(k+p-1) +\ldots+ E_{p-1}e(k+1).$$

Remark: Similarly, the implicit prediction scheme as well as the scheme for prediction by recursions through the process model (as in Sec.5.4) may also be developed for a multivariable process.

APPPENDIX 6

THE COVARIANCE MATRIX FOR p-STEP PREDICTOR

The covariance matrix $\mathbf{P}(k+p|k)$ is indicative of the degree of confidence that the estimator has in $\hat{\mathbf{x}}(k+p|k)$, the p-step ahead prediction of the state $\mathbf{x}(k)$. The derivation of $\mathbf{P}(k+p|k)$ is presented in this appendix.

Since the multiple prediction (6.7.5) is obtained by recursion through the time-update stage alone, the prediction error is largely dependent on $\mathbf{P}(k|k)$ and the process noise covariance $\mathbf{Q}$.

The prediction error sequence,

$$\{\tilde{\mathbf{x}}(k+p|k),\ p = k+1,\ k+2,\ \ldots\},$$

is a Gauss-Markov process as shown below. Rewriting (6.7.6), the prediction error

$$\tilde{\mathbf{x}}(k+p|k) = \mathbf{A}^{p}\tilde{\mathbf{x}}(k|k) + \sum_{i=k}^{m}\mathbf{A}^{m-i}\mathbf{S}\mathbf{w}(i), \qquad (6A.1)$$

where $m = k+p-1$.

Since the zero mean noise $\mathbf{w}$ possesses the property (6.6.5) the process $\{\tilde{\mathbf{x}}(k+p|k)$ in (6A.1) is a zero mean discrete-time Gaussian process:

$$\tilde{\mathbf{x}}(k+p-1|k) = \mathbf{A}^{p-1}\hat{\mathbf{x}}(k|k) + \sum_{i=k}^{m}\mathbf{A}^{m-i}\mathbf{S}\mathbf{w}(i).$$

Hence

$$\tilde{\mathbf{x}}(k+p|k) = \mathbf{A}\tilde{\mathbf{x}}(k+p-1|k) + \mathbf{S}\mathbf{w}(k+p-1), \qquad (6A.2)$$

which is a Markov property and hence $\{\tilde{\mathbf{x}}(k+p|k)\}$, the prediction error process is a Gauss-Markov process.

The prediction error covariance matrix

$$\mathbf{P}(k+p|k) = E[\tilde{\mathbf{x}}(k+p|k)\tilde{\mathbf{x}}(k+p|k)^{T}]. \qquad (6A.3)$$

From (6A.1) and (6A.3),

$$\begin{aligned}\mathbf{P}(k+p-1|k) = {} & E[\mathbf{A}^{p}\tilde{\mathbf{x}}\ (k|k)(\mathbf{A}^{p}\tilde{\mathbf{x}}(k|k))^{T}] \\ & + E[\sum_{i=k}^{m}\mathbf{A}^{m-i}\mathbf{S}\mathbf{w}(i)(\sum_{i=k}^{m}\mathbf{A}^{m-i}\mathbf{S}\mathbf{w}(i)^{T}] \\ & + E[\mathbf{A}^{p}\tilde{\mathbf{x}}(k|t)(\sum_{i=k}^{m}\mathbf{A}^{m-i}\mathbf{S}\mathbf{w}(i))^{T}] \\ & + E[\sum_{i=k}^{m}\mathbf{A}^{m-i}\mathbf{S}\mathbf{w}(i)(\mathbf{A}^{p}\tilde{\mathbf{x}}(k|k))^{T}]. \qquad (6A.4)\end{aligned}$$

The last 2 terms in (6A.4) will disappear, because

$$E[\tilde{\mathbf{x}}(k|k)\mathbf{w}^T(i)] = E[\mathbf{x}(k)\mathbf{w}^T(i)] - E[\hat{\mathbf{x}}(k|k)\mathbf{w}^T(i)], \tag{6A.5}$$

where

$$E[\mathbf{x}(k)\mathbf{w}^T(i)] = 0, \qquad i \geq k,$$

following (6.28b), and $\hat{\mathbf{x}}(k|k)$ can be expressed as a linear combination of output measurement, which is uncorrelated with the noise, and hence

$$E[\hat{\mathbf{x}}(k|k)\mathbf{w}^T(i)] = 0.$$

Using (6.6.5), (6A.4) can be expressed as

$$\mathbf{P}(k+p|k) = \mathbf{A}^p E[\tilde{\mathbf{x}}(k|k)\tilde{\mathbf{x}}(k|k)^T](\mathbf{A}^p)^T + \sum_{i=k}^{m} \mathbf{A}^{m-i}\mathbf{S}\, E[\mathbf{w}(i)\mathbf{w}(i)^T]\mathbf{S}^T(\mathbf{A}^{m-i})^T,$$

that is

$$\mathbf{P}(k+p|k) = \mathbf{A}^p\mathbf{P}(k|k)(\mathbf{A}^p)^T + \sum_{i=k}^{m} \mathbf{A}^{m-i}\mathbf{S}\mathbf{Q}(i)\mathbf{S}^T(\mathbf{A}^{m-i})^T.$$

APPENDIX 7A

DETAILS OF SELECTED EXAMPLES OF CHAPTER 7

The supportive details of the example presented in Sec.7.5.2 and Example 7.7(2) are given in this appendix.

7A.1 The Electrical Power Load Problem: Sec.7.5.2

It is assumed that 24-hourly data for 10 consecutive Mondays are available; here the first 10 sets of data from Appendix 7D are used for modelling. The objective is to produce prediction of the load for the following Monday.

The available 10x24 data set is appended with 8 columns of 'zeros to form the 10x32 data matrix **A**, which is WH-transformed to $\mathbf{A}_w$. The values of the cumulative square of each element of each column of $\mathbf{A}_w$ is computed, which are as follows (for column-1 to column-32):

(365165.5, 5.8, 15.2, 31.7, 72.5, 51.4,
64.8, 42.3, 29848.9, 7.6, 75.2, 57.9,
158.0, 9.0, 30.4, 136.5, 29049.9, 9.4,
85.0, 27.2, 142.2, 62.9, 256.0, 188.8,
68240.6, 10.9, 21.9, 17.5, 64.9, 182.0,
339.8, 70.0) x 10^3.

The series of elements each of the relatively dominant 11 columns of $\mathbf{A}_w$ are modelled using (7.5.1) and the estimated parameters are given by Table 7A.1.

Table 7A.1 Estimated parameters

Column	Parameters of series of column elements		
No.	f_0	f_1	f_2
1	3191.580	0.660	-0.189
9	590.266	0.920	-0.265
17	2289.569	0.162	-0.500
25	-2384.004	0.092	-0.011
13	-190.700	-0.025	-0.533
16	172.738	-0.517	0.041
21	80.044	-0.057	0.350
23	114.952	0.771	-0.516
24	-232.961	-0.470	-0.251
30	121.936	0.754	-0.638
31	288.721	0.145	-0.754

For each of the 11 columns of $\mathbf{A}_w$, the 11th column element is predicted using (7.5.2). For the other columns of $\mathbf{A}_w$, the predicted 11th element is assumed to be the same as the 10th element. The predicted 11th row of $\mathbf{A}_w$ so formed is reverse transformed to produce the prediction of the load of the 11th Monday, which is shown in Fig.7.5.1.

7A.2 Air Traffic Problem: Example 7.7(2)

This series is on the number of Trans-Atlantic Airline passengers (in thousands), is taken from Brown (1963, p.429).

The monthly data for the 12 consecutive years (1949 - 1960) are arranged in the 12 consecutive rows of matrix X:

$$\mathbf{X} = \begin{bmatrix} 112 & 118 & 132 & 129 & 121 & 135 & 148 & 148 & 136 & 119 & 104 & 118 \\ 115 & 126 & 141 & 135 & 125 & 149 & 170 & 170 & 158 & 133 & 114 & 140 \\ 145 & 150 & 178 & 163 & 172 & 178 & 199 & 199 & 184 & 162 & 146 & 166 \\ 171 & 180 & 193 & 181 & 183 & 218 & 230 & 242 & 209 & 191 & 172 & 194 \\ 196 & 196 & 236 & 235 & 229 & 243 & 264 & 272 & 237 & 211 & 180 & 201 \\ 204 & 188 & 235 & 227 & 234 & 264 & 302 & 293 & 259 & 229 & 203 & 229 \\ 242 & 233 & 267 & 269 & 270 & 315 & 364 & 347 & 312 & 274 & 237 & 278 \\ 284 & 277 & 317 & 313 & 318 & 374 & 413 & 405 & 355 & 306 & 271 & 306 \\ 315 & 301 & 356 & 348 & 355 & 422 & 465 & 467 & 404 & 347 & 305 & 336 \\ 340 & 318 & 362 & 348 & 363 & 435 & 491 & 505 & 404 & 359 & 310 & 337 \\ 360 & 342 & 406 & 396 & 420 & 472 & 548 & 559 & 463 & 407 & 362 & 405 \\ 417 & 391 & 419 & 461 & 472 & 535 & 622 & 606 & 508 & 461 & 390 & 432 \end{bmatrix}.$$

A 4x12 data window **A**(k) is assumed to move over **X**. So **A**(1) is composed of the 1st 4 rows of **X**, **A**(2) is composed of the 2nd to 5th rows of **X** and so on. SVD of **A**(1) to **A**(9) are computed. The singular values are shown in Table 7A.2.

Table 7A.2 Singular values ($s_i(k)$) for **A**(1) to **A**(9)

$s_i(k)$	k								
	1	2	3	4	5	6	7	8	9
s_1	1120.1	1294.5	1461.3	1664.7	1902.2	2160.7	2399.4	2650.4	2914.3
s_2	20.9	33.8	34.1	41.7	41.8	23.8	39.6	32.7	40.1
s_3	15.9	15.7	27.7	27.9	16.5	13.7	17.2	30.0	31.5
s_4	11.7	15.0	15.0	10.6	10.6	11.2	11.1	10.4	21.8

Reference

[1] Brown, R.G. (1963): *Smoothing, Forecasting, and Prediction of Discrete Time Series*, Prentice-Hall, Englewood Cliffs, New Jersey.

APPENDIX 7B

DATA ON OZONE COLUMN THICKNESS

The atmospheric ozone column thickness in the atmosphere measured at Arosa, Switzerland, is reproduced here from [1].

Table 7B.1 Monthly Ozone column thickness in Dobson units

Year	Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec
1931	---	---	---	---	---	---	---	311	335	283	286	301
1932	318	347	370	394	360	347	334	299	292	287	293	281
1933	357	364	399	382	390	374	335	319	309	312	311	337
1934	334	321	392	358	365	355	328	321	282	287	291	297
1935	332	390	367	383	375	319	331	311	288	275	299	313
1936	329	393	398	384	373	352	328	315	303	310	298	307
1937	347	352	395	382	365	349	324	323	301	283	280	355
1938	337	370	325	392	384	336	325	325	296	280	285	299
1939	320	341	385	347	382	339	331	313	286	304	284	309
1940	387	400	418	430	403	388	346	323	310	292	302	338
1941	362	395	417	409	417	361	348	336	306	299	309	309
1942	400	422	373	408	376	347	325	309	284	272	298	313
1943	338	341	385	363	348	352	336	303	291	292	303	321
1944	300	365	385	360	349	351	319	306	290	293	298	320
1945	377	359	360	373	376	351	329	327	297	288	295	313
1946	336	352	380	361	355	344	318	307	276	291	297	316
1947	383	397	393	369	361	347	334	324	307	296	278	312
1948	341	371	348	374	353	345	345	311	299	281	286	321
1949	332	365	378	357	371	354	335	321	284	272	296	292
1950	352	365	365	382	374	354	322	316	292	288	287	340
1951	338	402	417	397	383	364	332	321	297	298	278	311
1952	378	384	411	386	385	359	341	320	317	297	302	332
1953	335	375	373	383	382	359	326	317	293	280	266	---
1954	---	---	373	415	389	362	348	329	305	285	291	284
1955	315	375	399	374	361	351	339	332	300	292	278	317
1956	341	402	381	395	365	360	327	308	287	284	286	312
1957	340	342	353	375	380	349	330	321	305	279	294	322
1958	361	351	411	417	369	369	349	330	306	312	305	326
1959	369	367	364	390	389	373	342	330	313	294	297	328
1960	349	397	405	400	382	353	339	315	306	299	284	332
1961	368	333	338	365	379	349	343	322	296	288	301	304
1962	352	362	428	400	363	349	338	302	296	271	298	309
1963	371	408	377	381	378	352	327	310	282	273	278	293
1964	301	335	347	378	363	330	326	323	299	288	274	310
1964	332	390	383	385	387	345	337	318	305	275	278	305
1966	354	338	401	380	372	---	---	326	307	282	321	319
1967	361	362	354	374	349	358	325	317	296	273	267	304
1968	342	383	376	379	349	351	337	335	308	275	270	331
1969	327	419	361	393	351	364	333	333	294	281	307	345
1970	331	417	414	419	389	360	331	323	291	278	271	309
1971	344	349	411	364	358	358	336	309	317	278	292	306

Reference

[1] Bloomfield, P. (1985): 'Ozone column', in *Data*, D.F. Andrews and A.M. Herzberg (Eds.), Springer-Verlag, New York, 75-76.-76.

APPENDIX 7C

DATA ON ATMOSPHERIC CONCENTRATION OF CARBON DIOXIDE

The atmospheric concentration of carbon dioxide in parts per million for 22 consecutive years from 1959 to 1980 measured at the Mount Mauna Loa observatory in Hawaii is given in Table 7C.1; the data are presented row-wise for each year (from January to December) starting from 1959. The series is extracted from [1].

Table 7C.1 Atmospheric conc. of carbon dioxide in ppm.

Jan.	Feb.	March	April	May	June	July	August	Sept.	Oct.	Nov.	Dec.
315.16	315.97	316.37	317.40	317.96	317.82	316.23	314.54	313.60	313.03	314.57	315.32
316.10	316.68	317.37	318.79	319.63	319.29	317.86	315.55	313.85	313.64	314.61	315.81
316.54	317.34	318.12	319.06	320.20	319.44	318.24	316.52	314.57	315.13	315.75	316.73
317.70	318.29	319.37	320.25	320.84	320.43	319.35	317.13	316.01	315.19	316.42	317.47
318.45	318.82	319.72	321.06	321.87	321.22	319.44	317.48	315.89	315.83	316.72	317.98
319.17	---	---	---	322.08	321.92	320.42	318.58	316.68	316.65	317.60	318.49
319.32	320.36	320.82	322.06	322.17	321.95	321.20	318.81	317.82	317.37	318.93	319.09
319.94	320.98	321.81	323.03	323.36	323.11	321.65	319.64	317.86	317.25	319.06	320.26
321.65	321.81	322.36	323.67	324.17	323.39	321.93	320.29	318.58	318.60	319.98	321.25
321.88	322.47	323.17	324.23	324.88	324.75	323.47	321.34	319.56	319.45	320.45	321.92
323.40	324.21	325.33	326.31	327.01	326.24	325.37	323.12	321.85	321.31	322.31	323.72
324.60	325.57	326.55	327.80	327 80	327.54	326.28	324.63	323.12	323.11	323.99	325.09
326.12	326.61	327.16	327.92	329 14	328.80	327.52	325.62	323.61	323.80	325.10	326.25
326.93	327.83	327.95	329.91	330.22	329.25	328.11	326.39	324.97	325.32	326.54	327.71
328.73	329.69	330.47	331.69	332.65	332.24	331.03	329.36	327.60	327.29	328.28	328.79
329.45	330.89	331.63	332.85	333.28	332.47	331.34	329.53	327.57	327.57	328.53	329.69
330.45	330.97	331.64	332 87	333.61	333 55	331 90	330.05	328.58	328.31	329.41	330.63
331.63	332.46	333.36	334.45	334 82	334.32	333.05	330.87	329.24	328.87	330.18	331.50
332.81	333.23	334.55	335.82	336.44	335.99	334.65	332.41	331.32	330.73	332.05	333.53
334.66	335.07	336.33	337.39	337.65	337.57	336.25	334.39	332.44	332.25	333 59	334 76
335.89	336.44	337.63	338.54	339.06	338.95	337.41	335.71	333.68	333.69	335.05	336.53
337.81	338.16	339.88	340.57	341.19	340.87	339.25	337.19	335.49	335.51	336.63	337.74

Reference

[1] Keeling, C.D., R.B. Bacastow, and T.P. Whorf (1982): 'Measurements of the concentration of Carbon Dioxide at the Mauna Loa Observatory, Hawaii', *Carbon Dioxide Review 1982*, Ed. W.C. Clarke, Oxford Univ. Press, Oxford, 377-384.

Appendix 7D

DATA ON ELECTRICAL POWER LOAD ON A SUBSTATION

The electrical power load on a substation for 25 consecutive Mondays of the year 1983 are presented here. The hourly data for arranged row-wise in the following table for the consecutive Mondays.

Table 7D.1 Power load in MWH

Hours:											
01	02	03	04	05	06	07	08	09	10	11	12
139	131	122	123	139	157	188	239	256	265	213	229
152	149	143	147	156	166	199	258	272	271	256	241
148	144	141	142	148	170	208	257	265	258	255	223
144	139	135	133	140	162	195	262	270	270	255	238
158	146	140	141	144	165	211	270	279	282	270	257
154	150	145	151	157	171	215	257	278	273	254	242
163	154	152	152	155	175	222	251	279	269	269	249
164	156	153	143	161	180	207	250	278	276	260	239
178	169	166	112	172	183	221	256	270	266	262	252
199	195	180	188	180	196	223	256	269	269	279	262
205	194	185	176	187	199	229	251	267	282	263	253
211	196	186	187	191	201	234	247	285	272	333	279
223	213	207	196	199	212	225	249	259	281	274	271
227	219	216	211	213	209	216	222	230	223	219	210
223	217	206	197	199	203	211	238	268	272	283	271
228	209	212	205	200	213	213	233	266	274	287	276
239	230	225	203	202	205	208	230	256	276	277	281
235	228	218	209	210	215	217	246	258	287	282	286
229	213	211	209	208	211	220	230	261	276	252	278
225	217	214	196	205	210	209	219	245	273	279	275
220	213	207	203	203	207	210	226	252	276	292	272
210	206	197	211	214	213	216	228	253	277	277	274
220	217	215	214	212	215	210	219	252	275	275	272
214	209	205	204	202	209	218	234	257	289	289	279
210	200	199	197	199	209	224	253	279	290	285	279

(Contd.)

Table 7D.1 Power load in MWH (Contd.)

Hours:											
13	14	15	16	17	18	19	20	21	22	23	24
218	205	195	198	197	214	253	237	273	245	199	158
220	209	208	211	218	213	262	297	272	254	220	180
214	206	201	205	209	209	255	279	274	254	207	167
217	208	209	218	220	218	256	281	276	254	220	186
226	223	228	225	217	224	253	288	286	270	217	185
220	223	227	229	230	228	261	283	279	266	223	192
234	222	222	218	223	217	259	299	283	273	235	192
221	215	211	208	212	207	239	287	266	247	199	174
234	237	240	237	233	226	251	302	294	283	248	203
245	247	246	249	243	230	262	305	297	294	263	225
239	248	241	239	226	237	251	289	302	280	265	226
256	259	273	272	272	256	260	307	315	299	256	234
256	257	263	282	253	248	248	320	306	288	286	237
212	197	209	199	196	194	215	267	283	275	253	219
259	261	264	266	254	241	252	302	298	297	283	254
266	265	273	276	272	247	267	305	311	320	281	257
265	266	278	264	265	241	248	303	310	300	284	259
292	243	275	274	266	242	242	302	310	311	280	251
273	260	268	266	259	247	212	296	313	307	276	251
255	265	266	267	258	244	243	292	304	307	277	239
263	268	270	256	253	232	229	288	297	297	267	241
264	263	267	264	253	236	236	257	302	296	276	242
269	253	268	271	265	240	231	273	299	294	275	237
275	268	278	273	257	260	262	298	306	306	275	232
257	256	264	264	261	240	258	300	318	281	265	234

APPENDIX 7E

DATA ON UNEMPLOYMENT IN GERMANY

The monthly figures on the number of people unemployed in Germany during the period 1948 to 1978 are given below. The data have been reproduced from [1]. The monthly data for each year are presented row-wise.

Table 7E.1 Unemployment figures in Germany

Jan	Feb.	March	April	May	June	July	August	Sept.	Oct.	Nov.	Dec.
481971	476353	471803	469382	446943	451091	665016	784232	784126	739423	715128	759623
962866	1068885	1168127	1232381	1256889	1383302	1302857	1308091	1313691	1316572	1383832	1558469
2200486	2288368	2155962	2074220	1942134	1808534	1739507	1635604	1566588	1508348	1595491	1976461
2113553	1948422	1850960	1736166	1673661	1611908	1584067	1543866	1502799	1476741	1570796	1931002
2106836	2172973	1848101	1728250	1602178	1534867	1431499	1372614	1309563	1276009	1496764	1955635
2081227	2060651	1631613	1479538	1400709	1312201	1237767	1186715	1148914	1169558	1331378	1747757
2217243	2275347	1629817	1473474	299533	1198475	1109003	1042697	982184	977389	1117323	1464489
1975159	2000102	1578827	1047886	876414	790579	692089	630088	610855	627627	728116	1185850
1390335	1982469	1158631	754433	653274	586675	529246	503216	501687	516321	744959	1202533
1601562	1222789	804115	690039	587952	544987	467085	435506	436205	435751	557143	1320321
1533557	1418192	1201913	678161	556518	481151	422220	393066	387484	421322	497959	1022634
1445508	1203311	667087	466799	386004	314389	258048	235253	223206	235428	273606	512409
684297	581440	297704	225051	184564	162558	140701	132931	130861	142001	158424	302812
422744	321910	187840	153046	130387	115126	107704	111304	107883	112935	127792	239975
286398	273789	205467	135430	109403	97466	93939	91344	91383	101984	131356	232653
410047	416889	216323	143659	124037	112083	106111	104243	104507	114335	133024	252329
337497	304690	227188	146634	126654	112166	105399	102835	100266	111462	126844	202086
286334	291236	200978	126862	106541	95419	89018	85677	84974	92231	118962	177908
268848	235816	141428	121288	107743	100697	101476	105743	112726	145804	216382	371623
621156	673572	576047	501303	458461	400773	377235	359473	341078	360846	395004	526218
672617	589707	459853	330851	264674	226552	202689	187778	174467	180223	196056	266372
368585	374124	243212	155181	122967	110744	108018	103753	100477	107770	118849	192174
286266	264080	197784	120550	103407	94767	98562	99460	97338	110849	129476	175058
286171	254753	206472	160356	142890	135157	141975	145835	146740	170111	207990	269810
375564	368952	268461	231219	208289	190224	196774	198266	194660	214880	235379	279237
356352	347053	286576	240734	211276	200950	216616	221905	219271	266969	331839	485631
620494	620154	561762	517365	456965	450684	490894	527051	556981	672312	799337	945916
1154295	1183501	1114048	1087078	1017716	1002135	1035235	1031122	1005495	1061128	1114190	1223396
1350990	1346723	1190159	1093693	953538	921037	944609	939528	898701	943685	984699	1089935
1248918	1213741	1084229	1039228	946498	930974	927624	963468	911239	954376	1004325	1090708
1213498	1224309	1098969	1000429	912997	877319	922230	923963	864274	901636	927043	1006724
1171353	1134060	957711	875452	775117	763173	803653	798867	736809	761724	798973	866783
1036519	992520	875909	825374	766768	781396	853077	864519	822565	888100	967533	1118500

Reference

[1] Subba Rao, T., and M.M. Gabr (1984): *An Introduction to Bispectral Analysis and Bilinear Time Series Models*, Lecture Notes in Statistics, No.24, Springer-Verlag, New York,240-243.

APPENDIX 7F

DATA ON HOMOGENEOUS INDIAN RAINFALL

In India, the summer monsoon rainfall shows considerable spatial variability. The data presented here concern spatially coherent rainfall pattern over the north-western and central parts of Indian covering about 55% of the total area of the country. The monthly rainfall data at 14 meteorological sub-divisions over the years 1871-1990 have been used by Parthasarathy, Rupa Kumar and Munot (1993) to prepare the homogeneous rainfall data set. The data from 1940 to 1990 are extracted and presented here.

Table 7F.1 Monthly rainfall in mm

Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
11.0	10.9	10.2	11.1	21.4	146.8	274.5	268.6	66.1	49.3	13.6	7.6
17.8	11.1	6.1	3.2	10.5	90.5	218.0	175.9	96.6	19.2	0.7	1.7
11.9	30.7	1.6	10.6	9.9	137.0	360.4	275.8	130.5	7.8	0.7	14.7
32.2	1.8	1.2	9.1	32.1	122.4	300.7	137.7	187.7	64.7	3.9	0
12.4	23.2	43.3	9.0	8.1	96.9	375.4	320.8	111.1	64.2	6.6	1.3
19.9	1.0	0.6	19.6	8.4	147.2	331.5	212.3	202.1	28.6	3.0	0.3
---	11.1	1.1	13.3	16.1	192.7	251.3	291.5	102.4	17.5	64.8	10.9
15.4	10.4	6.6	14.8	6.7	62.3	256.2	309.7	236.2	24.6	3.0	11.7
27.4	9.2	6.6	7.9	5.9	113.6	253.3	233.1	161.3	27.9	65.1	1.0
1.4	5.8	1.3	4.6	31.0	95.2	279.7	184.8	228.3	62.7	1.9	0.4
4.5	12.6	15.6	1.6	9.7	60.6	352.3	170.7	195.4	18.9	4.9	4.8
4.3	1.8	22.6	14.5	17.4	100.8	222.9	194.4	82.8	49.2	14.3	0
1.2	9.7	5.4	7.9	17.4	117.8	289.0	207.7	71.2	28.4	0.2	3.0
15.6	1.2	0	12.3	2.5	113.4	252.3	302.1	132.2	55.9	0	0.6
5.6	12.8	7.1	5.5	6.3	95.4	285.6	193.0	279.1	22.5	0.3	2.8
13.9	1.3	3.7	7.1	17.1	156.0	160.9	333.9	221.9	119.0	3.0	0.2
4.9	3.6	6.2	4.2	44.1	139.5	388.8	235.6	135.3	102.2	29.3	4.6
14.8	2.6	31.3	14.4	18.6	105.4	245.2	261.8	81.4	38.6	8.9	2.1
4.4	6.0	7.8	11.0	14.7	86.4	308.0	244.9	214.1	59.5	19.6	3.1
12.7	3.6	0.8	8.9	21.5	123.8	357.7	255.4	234.8	76.4	9.8	0.3
13.1	0	19.0	5.5	20.7	131.4	232.6	234.7	98.8	44.6	4.5	3.0
12.8	13.1	2.2	8.2	30.7	143.4	328.1	249.0	265.5	85.5	5.9	4.5
4.3	7.1	11.9	18.3	20.0	55.4	258.5	194.1	208.3	20.8	13.7	32.9
1.9	5.6	8.7	13.2	13.5	112.7	207.0	310.3	126.9	51.7	12.0	1.1

continued

Table 7F.1 Monthly rainfall in mm (contd.)

Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
0.7	3.0	3.1	4.4	10.6	125.5	267.8	275.8	164.4	36.3	5.5	1.3
7.3	3.2	7.1	10.7	8.8	67.8	270.4	148.7	110.9	8.7	2.4	7.4
9.2	4.8	3.9	4.6	26.8	102.4	232.3	157.8	122.9	11.4	27.8	10.0
0.8	0.5	53.1	7.6	5.9	124.4	297.3	230.4	145.2	16.4	1.6	53.7
11.6	12.1	16.1	9.6	5.5	69.9	256.1	169.5	113.8	31.2	4.8	2.1
3.8	2.1	2.5	6.0	19.2	75.2	288.1	211.5	167.5	14.8	24.1	2.5
13.8	12.3	10.7	6.7	21.1	196.3	192.6	330.0	187.8	23.6	0.1	0
7.5	5.7	5.2	9.7	38.0	196.0	225.4	210.6	137.0	55.9	0.3	0
1.0	5.8	0.8	6.4	6.7	96.8	150.0	215.8	73.2	15.3	21.7	1.7
2.2	9.4	1.0	2.5	10.5	85.3	294.8	333.8	164.1	79.9	2.4	4.2
0.3	2.8	2.3	4.9	31.9	72.3	205.2	203.6	82.0	94.3	3.3	2.8
5.9	7.3	7.9	1.8	10.5	149.4	260.0	273.1	228.7	100.8	2.6	0.1
6.2	4.3	3.9	10.1	11.8	125.3	287.3	273.3	125.1	3.1	39.2	0.8
6.7	3.0	4.9	14.8	27.3	178.7	291.8	225.7	134.8	34.2	42.5	3.5
6.8	20.3	14.6	11.9	15.0	174.3	276.7	277.4	100.7	23.6	23.0	14.2
16.9	27.8	6.2	3.8	26.3	123.6	179.6	230.9	103.6	16.0	68.6	3.6
3.4	1.6	6.1	9.2	6.0	202.0	235.8	252.9	94.9	7.9	3.6	19.4
12.3	2.0	15.5	4.8	16.1	111.2	253.9	211.8	177.1	29.4	26.2	5.4
24.1	6.9	14.8	17.9	39.8	79.6	219.1	249.1	89.0	32.4	39.9	2.8

Reference

[1] Parthasarathy, B., K. Rupa Kumar, and A.A. Munot (1993): ‘Homogeneous Indian monsoon rainfall: variability and prediction’, *Proc. of Indian Acad. of Science (Earth Planet Sci.)*, 102(1), March, 121–155.

APPENDIX 8A

DATA ON YEARLY AVERAGED SUNSPOT NUMBERS

The count the number of spots on the sun's surface is of interest in Astronomy and Climatology for geo-physical reasons; the series is also of interest to time series analysts for the time-varying nature of the series. The daily observations from more than 50 observatories are used to arrive at the relative values of the sunspot numbers; the yearly averaged values for the years 1700 to 1987 are presented here.

The data are arranged column-wise, starting from the value for the year 1700.

Table 8A.1 Yearly averaged sunspot numbers

5	73	47.6	132.0	41.1	61.5	17.0	42.0	5.7	37.5
11	47	54.0	130.9	30.1	98.5	11.3	63.5	8.7	27.9
16	35	62.9	118.1	23.0	124.7	12.4	53.8	36.1	10.2
23	11	85.9	89.9	15.6	96.3	3.4	62.0	79.7	15.1
36	5	61.2	66.6	6.6	66.6	6.0	48.3	114.7	47.0
58	16	45.1	60.0	4.0	64.5	32.3	43.9	109.6	93.8
29	34	36.4	46.9	1.8	54.1	54.3	18.6	88.8	105.9
20	70	20.9	41.0	8.5	39.0	59.7	5.7	67.8	105.5
10	81	11.4	21.3	16.6	20.6	63.7	3.6	47.5	104.5
8	111	37.8	16.0	36.3	6.7	3.5	1.4	30.6	66.6
3	101	69.8	6.4	49.6	4.3	52.2	9.6	16.3	68.9
0	73	106.1	4.1	64.2	22.7	25.4	47.4	9.6	38.0
0	40	100.8	6.8	67.0	54.8	13.1	57.1	33.2	34.5
2	20	81.6	14.5	70.9	93.8	6.8	103.9	92.6	15.5
11	16	66.5	34.0	47.8	95.8	6.3	80.6	151.6	12.55
27	5	34.8	45.0	27.5	77.2	7.1	63.6	136.3	27.48
47	11	30.6	43.1	8.5	59.1	35.6	37.6	134.7	92.66
63	22	7.0	47.5	13.2	44.0	73.0	26.1	83.9	155.28
60	40	19.8	42.2	56.9	47.0	85.1	14.2	69.4	154.65
9	60	92.5	28.1	121.5	30.5	78.0	5.8	31.5	140.38
28	809	154.4	10.1	138.3	16.3	64.0	16.7	13.9	116.29
26	834	125.9	8.1	103.2	7.3	41.8	44.3	4.4	66.63
2	477	84.8	2.5	85.7	37.6	26.2	63.9	38.0	45.85
1	478	68.1	0.1	64.6	74.0	26.7	69.0	141.7	17.94
21	307	38.5	1.4	36.7					

APPENDIX 8B

DATA ON VARIATIONS IN THE ROTATION RATE OF EARTH

The variations in the rate of rotation of the Earth, are reproduced from [1]. The series has been of interest for the possible relationship with the sunspot numbers (Appendix 8A) and in general with the planetary system. Yearly data for the period 1820 to 1970 are presented here. The measurements are in 10^{-5}th of seconds.

Table 8B.1 Variations in the Earth's rate of rotation

1821	-217	1851	55	1881	-30	1911	361	1941	141
1822	-177	1852	51	1882	-12	1912	328	1942	150
1823	-166	1853	40	1883	11	1913	296	1943	157
1824	-136	1854	30	1884	57	1914	282	1944	143
1825	-110	1855	14	1885	92	1915	269	1945	138
1826	-95	1856	1	1886	86	1916	256	1946	137
1827	-64	1857	1	1887	53	1917	225	1947	151
1828	-37	1858	-4	1888	26	1918	202	1948	151
1829	-14	1859	-13	1889	6	1919	193	1949	136
1830	-25	1860	-56	1890	-12	1920	205	1950	111
1831	-51	1861	-83	1891	-35	1921	201	1951	105
1832	-62	1862	-104	1892	-31	1922	178	1952	105
1833	-73	1863	-93	1893	0	1923	139	1953	110
1834	-88	1864	-88	1894	36	1924	130	1954	104
1835	-113	1865	-75	1895	54	1925	101	1955	92
1836	-120	1866	-80	1896	65	1926	67	1956	96
1837	-83	1867	-101	1897	104	1927	22	1957	115
1838	-33	1868	-156	1898	166	1928	2	1958	144
1839	-19	1869	-226	1899	248	1929	12	1959	126
1840	21	1870	-293	1900	318	1930	26	1960	131
1841	17	1871	-333	1901	384	1931	21	1961	112
1842	44	1872	-347	1902	415	1932	10	1962	119
1843	44	1873	-329	1903	421	1933	-11	1963	139
1844	78	1874	-279	1904	402	1934	-12	1964	183
1845	88	1875	-205	1905	392	1935	-15	1965	206
1846	122	1876	-131	1906	387	1936	6	1966	231
1847	126	1877	-86	1907	391	1937	22	1967	244
1848	114	1878	-59	1908	396	1938	51	1968	239
1849	85	1879	-48	1909	400	1939	78	1969	263
1850	64	1880	-35	1910	391	1940	111	1970	273

Reference

[1] Luo Shi-fang, L. Shi-guang, Y. Shu-hua, Y. Shao-zhong, and L. Yuan-xi (1977): 'Analysis of periodicity in the irregular rotation of the earth', *Chinese Astronomy*, 1, 221-227.

APPENDIX 9

DATA ON COD PROCESS IN THE OSAKA BAY

The chemical oxygen demand (COD) can be considered to be an index of water pollution in the sea. COD concentration is monitored at a number of stations in the Osaka bay along with water temperature, transparency and dissolved oxygen concentration. Altogether 84 sets of monthly data are available, corresponding to the years 1976 to 1983.

The output variable is the COD concentration (y), which is related to the input variables: water temperature (x_1), water transparency (x_2) and dissolved oxygen concentration (x_3). y_f are the observed values of filtered COD (i.e. COD values of sea water free from suspended materials).

Table 9A.1 COD process data

Year: month	x_1	x_2	x_3	y_f	y	Year: month	x_1	x_2	x_3	y_f	y
1976:						1978:					
04	14.9	2.0	9.3	3	3.8	01	10	3	8.6	1.1	3.1
05	16.6	1.7	5.8	3.3	3.7	02	8.7	1.9	8.6	1.8	3.7
06	21.3	2.1	9.1	2.2	3	03	7.5	1.5	9.8	1.6	2.9
07	24.3	2.9	7	3.4	4.7	04	12.5	1.7	10	2.2	4.7
08	26.6	1.7	4.8	2.9	4.1	05	17.7	1.1	11	1.5	5.4
09	23.2	1.3	4.7	1.6	3.1	06	21.4	1.9	8.6	2.8	5.5
10	22.2	2.5	4.5	1.3	3.1	07	26.3	2.3	4.8	2.5	3.7
11	18.1	2.5	5.9	2	2.5	08	28.7	1.1	8.2	3.8	8.1
12	13.7	2.8	7.9	3.3	3.3	09	26.7	2.5	8.6	2	4.1
1977:						10	23.2	2.5	5.4	1.6	2.9
01	7.7	2.6	9.3	0.8	2.3	11	19.1	3	5.1	1.7	3.1
02	6.9	2.3	10.3	1.5	3.3	12	14.2	3	7.1	2.1	2.3
03	7.4	1.9	9.4	3.7	3.8	1979:					
04	11.3	2	8.4	2.2	3	01	12.5	2.5	7.4	1.8	2.9
05	17.6	2	8.2	2.4	2.7	02	9.3	2.4	9.4	2.2	2.8
06	19.5	2.4	6.5	1	3.3	03	9.5	2.5	9.4	2.1	2.7
07	21.6	2.5	5.1	1.3	4.9	04	12.5	2.5	9.1	2.1	4.3
08	26.5	1.9	6.3	1.8	4.8	05	16.0	2.1	10.0	2.9	4.9
09	26.4	2.2	8.1	2.8	4.4	06	19.7	1.6	10.0	3.0	5.9
10	23.1	3.8	4.9	2.2	2.2	07	22.8	3.0	5.0	2.0	3.2
11	21.1	2.5	5.1	0.8	1.8	08	28.2	1.0	10.0	3.3	8.4
12	15.2	2.5	6.5	1.2	2.9	09	24.6	2.8	6.4	2.1	4.3

(Contd.)

Table 9A.1 COD process data (contd.)

Year: month	x_1	x_2	x_3	y_f	y
1979:					
10	22.2	2.3	5.9	2.6	3.6
11	17.9	2.5	6.7	2.0	3.0
12	14.3	4	7.5	1.5	3.1
1980:					
01	7.3	2.0	9.4	2.5	3.4
02	6.6	2.0	11	3.0	4.6
03	8.8	2.3	8.7	3.0	3.8
04	13.6	2.7	10	2.7	3.6
05	13.3	2.5	7.6	5.1	5.6
06	18.6	1.9	6.8	3.9	5.1
07	23.6	2.0	9.1	3.0	4.2
08	26.9	2.1	7.2	4.6	8.2
09	25.0	3.0	4.2	2.7	3.5
10	22.1	3.0	5.8	3.6	6.9
11	18.2	5.0	6.7	3.7	3.8
12	16.6	3.5	6.0	2.0	4.0
1981:					
01	5.1	2.7	10	3.5	3.8
02	6.7	2.0	10	2.9	3.9
03	7.5	2.5	9.0	2.6	2.8
04	13.1	2.7	8.8	1.3	1.9
05	14.7	1.8	5.1	2.3	3.2
06	17.1	1.5	3.7	1.7	2.7
07	19.8	1.0	1.8	1.9	2.1
08	23.9	1.0	1.5	0.8	2.0
09	25.8	2.0	4.6	1.4	2.0
10	23.0	2.0	6.3	1.2	2.5
11	19.6	2.8	6.7	1.2	1.9
12	15.2	2.5	7.1	1.0	1.6
1982:					
01	8.5	2.5	8.4	1.8	3.4
02	8.1	4.0	8.5	1.3	1.7
03	8.2	2.6	8.7	1.3	2.1
04	14.8	2.0	10	2.1	5.3
05	17.5	1.5	11	3.2	6.8
06	19.3	2.7	6.5	2.6	4.2
07	23.1	2.1	6.2	3.2	6.3
08	26.1	1.2	9.4	3.9	7.2
09	23.1	2.5	3.7	2.8	3.1
10	21.7	2.3	6.1	1.8	3.0
11	19.5	2.5	5.5	1.8	2.9
12	14.3	3.6	7.0	2.0	2.3
1983:					
01	8.6	3.3	8.4	1.9	2.3
02	8.5	3.5	10	2.1	2.5
03	8.3	3.0	9.1	1.7	1.9

References

Remark: Data from 1976 to 1981 appear in [1]. The rest are obtained from [2].

[1] Shin-Ichi, Fujita and Hiroshi, Koi, (1984): 'Application of GMDH to Environmental System Modelling and Management' in *Self-organizing Methods in Modelling*, Ed. S.J. Farlow, Marcel Dekker, New York.

[2] Shin-Ichi, Fujita (1991): Private communication.

APPENDIX 10

GENERALIZED DELTA RULE

The derivation of the generalized delta rule (GDR) which is due to Rumelhart, Hinton and Williams (1986) is presented here. GDR gives an expression for the adaptive change in the weights on the interconnection between the nodes (see Fig.10.2.2) minimizing the cost

$$J_z = \frac{1}{2}\sum_{j=1}^{N}(E_{zj})^2, \qquad (10A.1a)$$

with the error, E_{zj}, defined as

$$E_{zj} = (y_j - O_{zj}) \qquad (10A.1b)$$

where y_j and O_{zj} are the desired output and the computed output respectively of the jth node of the output layer **Z** which has N number of columns.

Define O_{Qj} and S_{Qj} as the nodal output and the sum of all weighted inputs respectively at the jth node of the Qth layer (Fig.10A.1). The inputs are O_{Pi}, and W_{ij} is the weight on the interconnection between i-th node of one layer and j-th node of next layer. Assume the layers from the output end are **Z**, **Y**, **X** etc., having N, M, L, etc. number of nodes respectively. F_{Qj} is the threshold level.

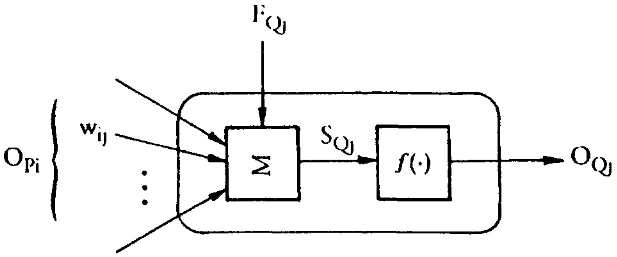


Figure 10A.1 Topology of a generic node Q_j

GDR is based on the gradient descent algorithm, according to which the adaptation in the weights W_{ij} is proportional to the gradient error; for example for the jth node of **Z** layer

$$\Delta W_{ij} \;\alpha\; -\frac{\partial J_z}{\partial W_{ij}} = -\frac{\partial J_{zi}}{\partial O_{zj}}\frac{\partial O_{zj}}{\partial S_{zj}}\frac{\partial S_{zj}}{\partial W_{ij}}. \qquad (10A.2)$$

$$S_{zj} = \sum_{i=1}^{M} W_{ij} O_{Yi} + F_{zj}$$

Hence

$$\frac{\partial S_{zj}}{\partial W_{ij}} = O_{Yi}. \tag{10A.3}$$

Again

$$O_{zj} = f_{zj}(S_{zj}),$$

and assuming $f(.)$ to represent sigmoidal nonlinearity:

$$f(x) = \frac{1}{1+e^{-x}},$$

$$\frac{\partial O_{zj}}{\partial S_{zj}} = \frac{\partial f_{zj}(S_{zj})}{\partial S_{zj}}$$

$$= \frac{e^{-s_{zj}}}{(1 + e^{-s_{zj}})^2} = \frac{1}{(1 + e^{-s_{zj}})}\left[1 - \frac{1}{(1 + e^{-s_{zj}})}\right]$$

$$= O_{zj}(1-O_{zj}). \tag{10A.4}$$

Again, following (10A.1),

$$\frac{\partial J_z}{\partial O_{zj}} = -(y_j - O_{zj}).$$

Hence using (10A.2), (10A.3) and (10A.4),

$$\Delta W_{ij} = \alpha(y_j-O_{zj})O_{zj}(1-O_{zj})O_{Yi},$$

where α $(0<\alpha<1)$ is the proportionality constant. Thus

$$\Delta W_{ij} = \alpha O_{Yi} D_{zj},$$

where

$$D_{zj} = (y_j- O_{zj})O_{zj}(1-O_{zj})$$

is the discrepancy or error corresponding to the node Z_j. Hence the error at the output of each node of layer Y due to the error D_{zj} for $j = 1$ to n:

$$\varepsilon_{Yi} = \sum_{j=1}^{N} W_{ij} D_{zj}.$$

So error at output of node Y_i,

$$J_Y = \sum_{i=1}^{M} \varepsilon_{Yi} O_{Yi}. \tag{10A.5}$$

Hence

$$\frac{\partial J_Y}{\partial O_Y} = \varepsilon_{Yi}.$$

Now it is intended to compute changes in the weights V_{hi} on the interconnections between X layer nodes and Y layer nodes.

$$\Delta V_{hi} \; \alpha \; \frac{\partial J_Y}{\partial V_{hi}} = \frac{\partial J_Y}{\partial O_{Yi}} \frac{\partial O_{Yi}}{\partial S_{Yi}} \frac{\partial S_{Yi}}{\partial V_{hi}}.$$

Using (10A.5), (10A.3) and (10A.4),

$$\begin{aligned}\Delta V_{hi} &= \beta\left(\sum_{j=1}^{N} W_{ij} D_{zj}\right) O_{Yi}(1 - O_{Yi}) O_{Xh} \\ &= \beta D_{Yi} O_{Xh}.\end{aligned}$$

β is the proportionality constant, and

$$D_{Yi} = \left(\sum_{j=1}^{N} W_{ij} D_{zj}\right) O_{Yi}(1 - O_{Yi})$$

is the discrepancy or error corresponding to node Y_i.

If the nodes have threshold level inputs, the change in the threshold levels for the Z_j nodes is given by

$$\Delta F_{zj} = \alpha D_{zj}$$

Similarly, for the Y_i node

$$\Delta F_{Yi} = \beta D_{Yi}.$$

The procedure continues to earlier layers, if any.

Remark: A modified version of the algorithm, referred to as *recurrent backpropagation algorithm* (Pineda, 1987) can be used for the training of recurrent networks.

References

[1] Rumelhart, D.E., G.E. Hinton, and R.J.Williams (1986): 'Learning internal representations by error propagation', in *Parallel Distributed Processing*, Vol.1, Ch.8, D.E. Rumelhart, and J.L. McClelland, Eds., M.I.T. Press, Cambridge, MA.

[2] Pineda, F.J. (1987): 'Generalization of backpropagation to recurrent neural networks', *Physical Review Letters*, 59, 2229-2232.

APPENDIX 11

SVR SPECTRUM

SVR spectrum is a method of determining the period length of periodic components present, if any, in any signal or data sequence; the periodic components need not be sinusoidal. The data $\{y(k)\}$ are arranged into the consecutive rows of a matrix (**A**) which is singular value decomposed; the generic term *Singular Value Ratio* (SVR) spectrum stands for the spectrum of a function (usually squared ratio) of the most dominant and other singular values against varying row lengths of the data matrix.

The singular value decomposition (SVD) of an mxn matrix **A** is given by

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \sum_{i=1}^{p} \mathbf{u}_i s_i \mathbf{v}_i^T, \qquad p = \min(m,n),$$

$\mathbf{u}_i$ and $\mathbf{v}_i$ being the column vectors of **U** and **V** respectively; the singular values (s_i) of **A** appear in nonincreasing order down the diagonal of **S** (see Sec.7.6). If the data series $\{y(k)\}$, which is contained in the consecutive rows of **A**, is perfectly periodic with period length n, the rows will be aligned with respect to each other, and **A** will be a rank-one matrix. Hence the first singular value s_1 will be large, whereas the other singular values will be zero. However if the row length of **A** is different from n, there will be misalignment between the rows and **A** will no longer be a rank-one matrix resulting in relatively lower value of s_1 with respect to the other singular values.

Hence the ratio,

$$\rho(n) = s_1^2/s_2^2, \tag{11A.1}$$

may be used as an index indicating the presence of a periodic component of period length n in a data sequence $\{y(k)\}$, where s_1 and s_2 are the first two singular values of the matrix **A** of row length n, which contains the data. The distribution of $\rho(n)$ against n is termed as the SVR spectrum. It has two prime features:

(i) The spectrum will show a peak at the period length (N) of the strongest periodic component present in the data sequence;

(ii) the spectrum will also show peaks at multiples of this period length (i.e. at N, 2N,...).

Remarks

(1) In the case of a dynamic series, the concept of an (overlapping) moving window may be considered (see Appendix 7A.2) for the formation of the data matrix **A**(K), where K is the window index. For each row length, the ratio s_1^2/s_2^2 for each **A**(K) is computed and the averaged (or median, i.e. centre of the serially arranged data set in order of magnitude) value is considerd as the $\rho(n)$ for constructing the SVR spectrum.

(2) In place of $\rho(n)$, an alternative expression in terms of the relative energy in the first decomposition component may be used which is given by

$$\rho_E(n) = \left(s_1^2 / \sum_{i=1}^{p} s_i^2\right).$$

where

$$\sum_{i=1}^{p} s_i^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2, \qquad \{\{a_{ij}\}\} = \mathbf{A},$$

A being an mxn matrix.

Summary for computation of SVR spectrum

(1) The data series is arranged into an mxn matrix **A**.
(2) **A** is SV-decomposed.
(3) $\rho(n) = s_1^2/s_2^2$ is computed.
(4) n is incremented to n+1, and steps 1 to 3 are repeated until m = 2 or n is too large to be of interest.
(5) The plot of $\rho(n)$ against n is the desired SVR spectrum.

Features

(1) If the discrete-time signal or data sequence is a sampled version of a continuous-time phenomenon, the data are more closely sampled and the detection of the period length by the SVR spectrum is more accurate.

(2) SVR spectrum requires only the singular values of **A**; the singular vector matrices **U** and **V** need not be computed.

(3) It is observed that high values of $\rho(n)$ are obtained for low values of n, where the series is almost linear. This initial monotonically decreasing part of the SVR spectrum should be ignored.

(4) In the case of noninteger period length of a periodic component, the peaks tend to drift at multiples of the length at which the first peak occurs. In such cases, the series may be expanded using equispaced interpolated data;

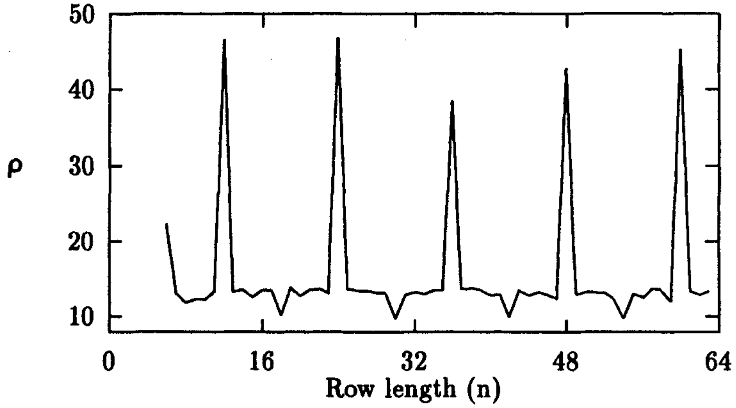


Figure 11A.1 SVR spectrum of the ozone column series.

SVR spectrum on expanded series is expected to produce closer detection of the period length of interest.

Examples

Three examples are presented as follows:

(a) The ozone column series (Appendix 7B): This is a monthly data series with yearly periodicity. The SVR spectrum (Fig.11A.1) shows repeating peaks at multiples of the period length 12, confirming the presence of a component of period length 12.
(b) A white Gaussian noise process: The SVR spectrum of this series (Fig.11A.2) shows absence of any dominent periodic component.
(c) A chaotic series {x(k)} generated by the Mackey-Glass equation:

$$x(k+1) - x(k) = \frac{\alpha x(k-\tau)}{1 + x^{\gamma}(k-\tau)} - \beta x(k),$$

with $\alpha = 0.2$, $\beta = 0.1$, $\gamma = 10$ and $\tau = 30$. This series is discussed in Sec.8.3.2. The SVR spectrum in Fig.11A.3 confirms the absence of any period component.

The use of SVR spectrum for decomposition of periodic components is discussed in Sec.11.5; the problem of the extraction of a signal from the composite signal using SVR spectrum is addressed in Sec.14.5.4.

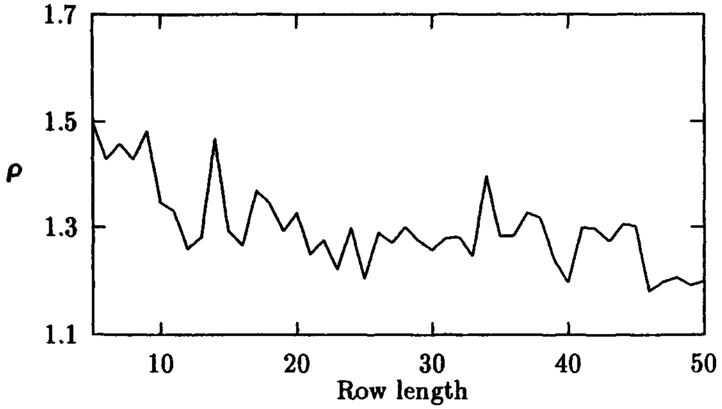


Figure 11A.2 SVR spectrum of white Gaussian noise.

The identification of the period length is based on closeness of rank-oneness; any feature of the series that influences the rank-oneness or near orthogonality of a component with respect to the rest, will affect the detection of period length. Although SVD is used in the present study, other orthogonal decompositions like the QR decomposition may be used for similar application.

SVR spectrum is a recently developed concept (Kanjilal and Palit, 1995); further research is expected to detail its prospects and limitations.

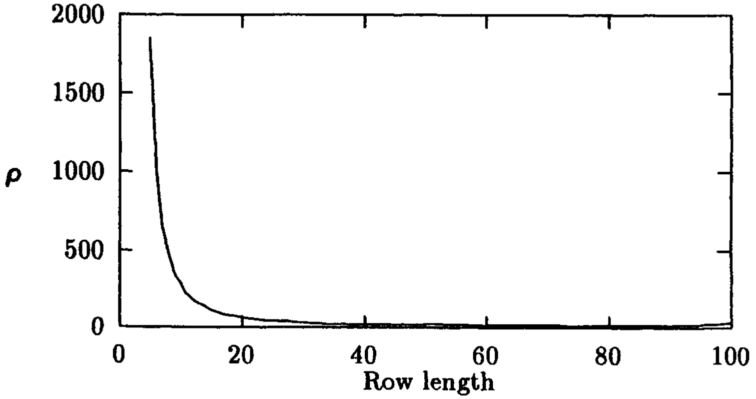


Figure 11A.3 SVR spectrum of the Mackey-Glass equation modelling a chaotic process.

Reference

[1] Kanjilal, P.P., and S.Palit (1995): 'On multiple pattern extraction using singular value decomposition', *IEEE Trans. on Signal Processing*, 43(6), 1536-1540.

APPENDIX 12A

SYSTEMS AND CONTROLS BASICS

Some basic concepts related to systems, models and control methods are introduced here.

Every system or process is, by itself, a continuous time process. However, for considerations related to measurement and computation, process measurements are often recorded at discrete-time intervals, and the monitored process (Fig.12A.1) is loosely called a discrete-time process.

Consider a discrete-time model

$$y(k) + a_1y(k-1) + a_2y(k-2) = b_ou(k-d) + b_1u(k-d-1) + b_2u(k-d-2),$$

where u is the input to the process and y is its output; the output responds to a change in control action after d sampling intervals. A concise expression using discrete-time polynomial operators follows as

$$A(q^{-1})y(k) = B(q^{-1})y(k-d),$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2},$$

$$B(q^{-1}) = b_o + b_1q^{-1} + b_2q^{-2}.$$

The process is also expressed as

$$y(k) = \frac{B(q^{-1})q^{-d}}{A(q^{-1})}\, u(k) = G(q^{-1})u(k),$$

where $G(q^{-1})$ is the transfer function of the process. Two

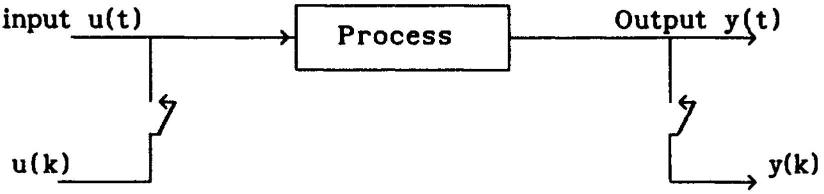


Figure 12A.1 Schematic of an open-loop process with u as input and y as the output; the process measurements are recorded at discrete-time intervals.

basic properties of a system are the nonminimum-phasedness and stability.

The process is called *open-loop stable*, if its poles (i.e. roots of $A(q^{-1})$) lie inside the unit circle; the converse is also true. The term open-loop implies the information of the process output is not available to the mechanism that generates the input to the process.

A discrete-time process is defined to be *nonminimum-phase*, if it has unstable zeros (i.e. roots of the numerator, $B(q^{-1})$ lying outside the unit circle in the z-plane) (Franklin and Powell, 1980). Such a process is also said to have an *unstable inverse*. Since a discrete-time process is basically obtained through the sampling of the continuous-time processes, the effect of sampling on the stability of the system deserves careful consideration. If a continuous-time process with a stable inverse is sampled too fast, the discrete-time process so generated can have an unstable inverse. Conversely, even if the continuous-time process has an unstable inverse, it can be sampled at a slow rate and the resulting discrete-time process can have a stable inverse. Again too slow a sampling rate may fail to capture fully the process dynamics and thereby lack in representativeness.

Example 12A Consider a discrete-time process with a transfer function

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} . \qquad (12A.1)$$

Consider two different sets of parameter for this process:

(a) $a_1 = -1.4$, $a_2 = 0.6$, $b_1 = 0.6$, $b_2 = -0.4$.

Here, the poles are at 0.7+j0.6633 and 0.7−j0.6633 and the zeros are at 0 and 0.66 (the unity time delay represents the zero at 0) in the z-plane.

(b) $a_1 = -1.4$, $a_2 = 0.6$, $b_1 = 0.4$, $b_2 = -0.6$.

Since the numerator has a root at 1.5, the process is a nonminimum-phase process in this case.

The response of the process (12A.1) to a unit step input (at time $k = 10$) for the two different parameter sets mentioned above is shown in Fig.12A.2. The process has unity steady-state gain (obtained for $q=1$ in (12A.1)). □□

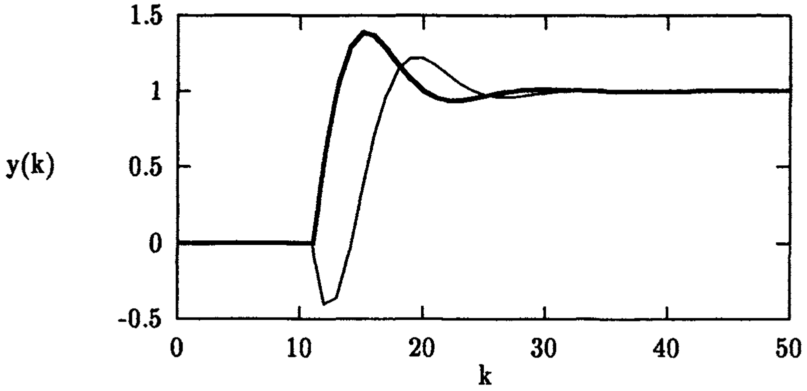


Figure 12A.2 Response of the system shown in Fig.12A.1 to unit step input:
——— with parameter set (12A.2(a)), and
——— with parameter set (12A.2(b)); the negative going output in the latter case is due to the nonminimum-phase nature.

A system or process with input u and output y is said to be *causal* if the output at any time is dependent on inputs up to that time; the output is also called the controlled variable. Causality is not a symmetric property, that is y can be causal output to input u but not vice-versa.

A system is said to be *time invariant* if the response of the system, i.e. the relationship between the output and the input does not vary with time. Again, a system is said to be *linear* if its inputs and outputs satisfy the principle of superposition; in other words, the output response of a linear time-invariant system due to a number of inputs is equal to the summation of the output responses corresponding to the individual inputs. For example, if the outputs y_1 and y_2 are obtained for the inputs u_1 and u_2 respectively to the systems the output response due to the input (u_1+u_2) will be (y_1+y_2).

The system, shown in Fig.12A.3, is said to be a *closed-loop* system, where the control input at any point in time is determined in consideration of the system response y up to that time. The controller is fed with the deviation of the

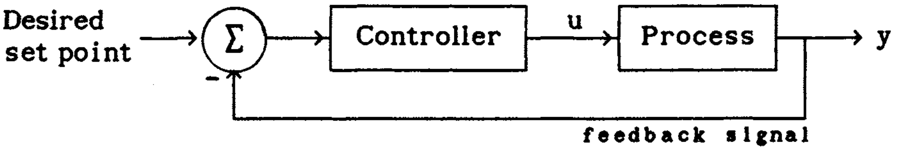


Figure 12A.3 Schematic diagram of a closed-loop system.

system response (used as the feedback signal) from the desired set point. In other words at every time k, the measurement of the system output is fed back to the controller (through some intermediate stages). To be precise, a control system is referred to as a *feedback* control system, if it utilizes the observations or measurements that are fed back but ignores the subsequent feedback information; a control system is said to be a *closed-loop* control system, if in addition to utilizing the feedback information on the process output, it is understood that the subsequent feedback information as well as the associated statistics will be available, i.e. the loop will stay closed.

The basic objective of process control is to make the process perform so that its output remains at the desired level. The type of control depends mainly on the nature of process dynamics. If the process characteristics do not vary with time, the process is called a *deterministic process*; in such cases the controller parameters will need to be tuned only once, or very infrequently.

It is necessary to resort to *adaptive control* when process dynamics vary with time. The two main units within an adaptive controller (Fig.12A.4) are the process (or controller) parameter estimator and the controller itself. Different designs are possible for the adaptive controller. If the change in the process dynamics is predictable from the available measurements or other information, *gain scheduling* can provide satisfactory control. In gain scheduling, the process parameters need not be estimated; here the controller parameters are in the form of a look-up table for various ranges of measurements over various modes of operations. The updating mechanism for the table of parameters based on controller performance may or may not be

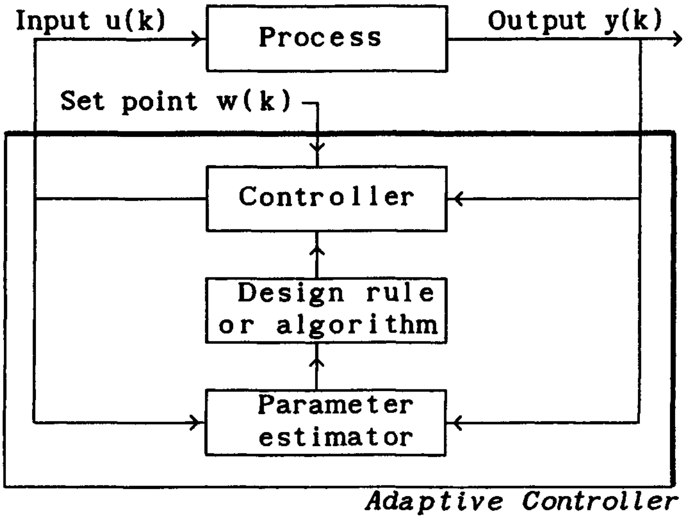


Figure 12A.4 Schematic diagram of a typical adaptive control system.

there. If the dynamics of the process vary so unpredictably that the controller parameters cannot be precalculated from the available measurements, the process (or controller) parameters are estimated and are used in the design of the controller. The self-tuning control, discussed in Sec.12.2, belongs to this category.

The controller should be optimal. The word *optimal*, by itself, is not very meaningful; a cost function has to be specified which the control algorithm has to minimize with respect to a process model. The quality of control performance will depend on the rationale of the cost function. In real life, optimal performance is difficult to achieve because real-life problems conform neither to the optimality criteria nor to the mathematical model of the process. However optimal control provides a guideline for the design of the controller and the idealized performance.

Most real-life processes are not deterministic and hence not exactly known. There is a factor of uncertainty in the mathematical model of the process. *Stochastic control* concerns control of processes in the presence of uncertainty. The control performance is not expected to be optimal, and hence the control is referred to as suboptimal control. *Suboptimal* is a mathematically undefined term, which is expected to mean 'close to optimal'. Adaptive control (Fig.12A.4), is a suboptimal control which incorporates (a) some method of estimation of the process parameters, and (b) a suitable control strategy to produce the control law in consideration of the latest process parameters.

One of the desirable properties of a controller is *robustness*. A robust control system is one that continues to perform in accordance with the design criteria, irrespective of the change in the system dynamics and hence in the system behaviour, from its mathematical model.

References

Remarks: This subject is nicely covered in [1,2].

[1] Aström, K.J., and B. Wittenmark (1984): *Computer Controlled Systems: Theory and Design*, Prentice-Hall, Englewood Cliffs, N.J.

[2] Franklin, G.F., and J.D. Powell (1980): *Digital Control of Dynamic Systems*, Addison-Wesley, Reading.

APPENDIX 12B

SMITH PREDICTOR

One of the fundamental works in predictive control is by Smith (1957), who designed a controller essentially free from the effects of the time delay.

A typical process shows an inherent time delay between the input u, and the process output y (Fig.12B.1). Let the time delay be expressed as G_d, given by $\exp(-s\tau)$, in continuous time, s being the Laplace operator. The control action can at best force the output to be equal to the set point in a time equal to the dead time of the process. Faster control performance is not possible. Under-estimation of the time delay and unnecessary control action can lead to instability.

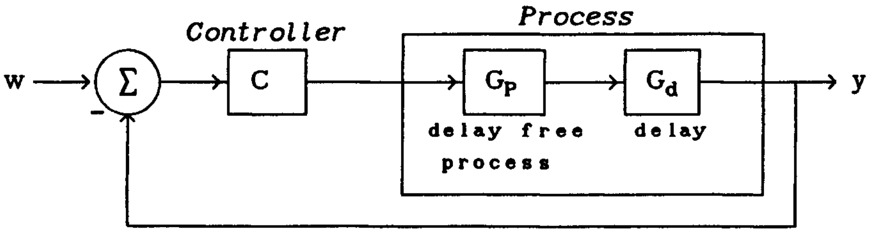


Figure 12B.1 A typical time-delay process.

To take account of the time delay of the process, instead of operating on the output error, $(w(t)-y(t))$, the controller should operate on the predicted output error, $(w(t)-y(t+\tau))$. The Smith predictor offers a novel approach for the realization of the output predictor, which with appropriate design of the controller results in a control performance free from the effects of time delay. This can be achieved by adding a negative feedback loop around the controller with $G_{pm}(1-G_{dm})$ as the feedback element, where G_{pm} and G_{dm} are the modelled process transfer function (G_p) and modelled time delay (G_d) respectively. The overall closed-loop realization is shown in Fig.12B.2. Using Laplace transforms, the effective controller C of Fig.12B.1, in the new configuration becomes C^*, where

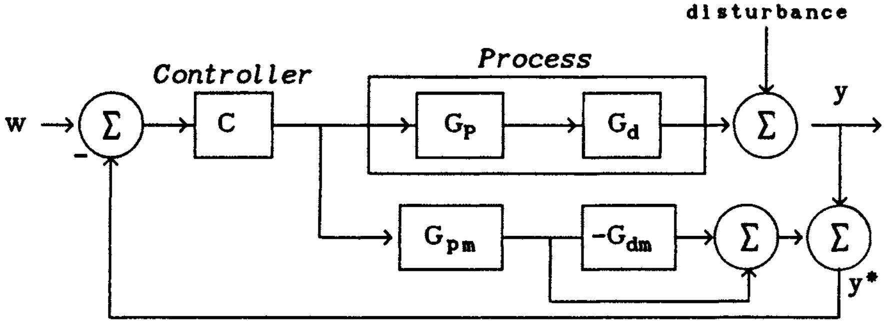


Figure 12B.2 The Smith predictor control scheme.

$$C^*(s) = \frac{C(s)}{1 + C(s)G_{pm}(s)(1 - \exp(-s\tau_m))}, \tag{12B.1}$$

and the overall closed-loop system (Fig.12B.2) is given by

$$\frac{y}{w} = \frac{CG_pG_d}{1 + CG_{pm} - CG_{pm}G_{dm} + CG_pG_d} \,. \tag{12B.2}$$

If $G_{pm} = G_p$ and $G_{dm} = G_d$, and if the disturbance is zero, (12B.2) reduces to

$$\frac{y}{w} = \frac{CG_pG_d}{1 + CG_p} \,. \tag{12B.3}$$

Note that (i) the closed-loop characteristic equation (12B.3) becomes free from the time-delay term, and (ii) y^* in Fig.12B.2 is effectively the prediction $y(t+\tau)$, if G_{pm} and G_{dm} are correctly modelled.

Some of the drawbacks of the Smith predictor scheme are as follows.

(a) A delay-free process model and a delay model are required. The scheme will not be able to stabilize open-loop unstable processes.

(b) The control performance is dependent on the modelling accuracy of G_{pm} and G_{dm}. The control performance is expected to be more sensitive to the drift in system parameters than its delay-free counterpart.

(c) The implementation of the modelled time delay can be difficult.

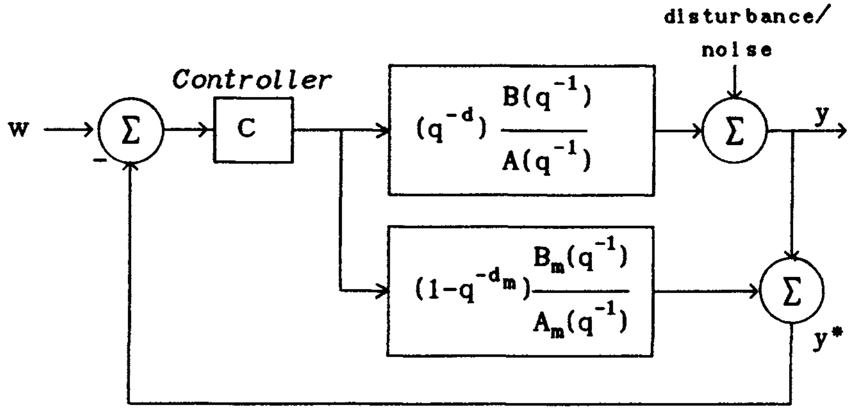

disturbance/
noise
Controller
w
Σ
-
C
$(q^{-d})\,\frac{B(q^{-1})}{A(q^{-1})}$
Σ
y
$(1-q^{-d_m})\frac{B_m(q^{-1})}{A_m(q^{-1})}$
Σ
y*

Figure 12B.3 The discrete-time Smith predictor control scheme.

If the discrete form of operation is permissible, most of the above drawbacks disappear. A discrete-time realization of the Smith predictor is shown in Fig.12B.3.

Here, $A(q^{-1})$ and $B(q^{-1})$ are polynomials in discrete time; q^{-1} is the unit backward shift operator, and d is the time delay in discrete sampling time intervals. The subscript m stands for modelled (or estimated) quantities.

Under certain conditions the Smith predictor control scheme can be shown to be equivalent to the self-tuning controller as shown in Gawthrop (1977). Here again y* can be interpreted as the d_m-step ahead prediction $\hat{y}(k+d_m|k)$, where d_m is the modelled time delay. The drifts in parameter variation can be easily taken care of by on-line parameter estimation.

For further discussions on the Smith predictor, see Marshall (1979).

References

[1] Gawthrope, P.J. (1977): 'Some interpretations of the self-tuning controller', *Proc. IEEE*, **124**, 10, 889-894.

[2] Marshall, J.E. (1979): *The Control of Time-delay Systems*, Peter Peregrinus, Stevenage.

[3] Smith, O.J.M. (1957), 'Closer control loops with dead time', *Chemical Engineering Progress*, 53(5), 217-219.

APPENDIX 13A

DERIVATION OF STATE-SPACE DETERMINISTIC LQ CONTROL

In this appendix the state-space formulation of the deterministic LQ control problem is presented. A multi-input multi-output process is considered. The derivation is based on dynamic programming, which was developed by Bellman in 1953 (Bellman and Dreyfus, 1962). The derivation can be easily simplified to the single-input single-output case.

Consider the deterministic model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \tag{13A.1}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \tag{13A.2}$$

where **x** is an nx1 state vector, **y** is rx1 vector of measured outputs, **u** is the mx1 vector of deterministic control inputs; **A**, **B**, and **C** are nxn, nxm, and rxn real matrices respectively which are assumed to be known. The initial state **x**(0) is known.

The objective is to produce the optimal control decisions **u**(k), **u**(k+1),..., **u**(k+N-1), so as to minimize the scalar cost function

$$J = \mathbf{x}^T(k+N)\mathbf{Q}_N\mathbf{x}(k+N) + \sum_{i=k}^{k+N-1}(\mathbf{x}^T(i)\mathbf{Q}_{N-1}\mathbf{x}(i) + \mathbf{u}^T(i)\Lambda\mathbf{u}(i)). \tag{13A.3}$$

where $\mathbf{Q}_N$ and $\mathbf{Q}_{N-1}$ are symmetric positive semidefinite matrices and Λ is a positive definite matrix.

Let the cost at the last stage of the horizon {k,k+N}, be defined as

$$J_N = \mathbf{x}^T(k+N)\mathbf{P}(k+N)\mathbf{x}(k+N), \tag{13A.4}$$

where $\mathbf{P}(k+N) = \mathbf{Q}_N$. Similarly the total cost over the last two stages (k+N-1 and k+N) is given by

$$J_{N-1} = \min_{\mathbf{u}(k+N-1)} \{\mathbf{x}^T(k+N-1)\mathbf{Q}_{N-1}\mathbf{x}(k+N-1) + \mathbf{u}^T(k+N-1)\Lambda\mathbf{u}(k+N-1) + J_N\}. \tag{13A.5}$$

From (13A.1),

$$\mathbf{x}(k+N) = \mathbf{A}\mathbf{x}(k+N-1) + \mathbf{B}\mathbf{u}(k+N-1). \tag{13A.6}$$

Substituting for J_N in (13A.5) using (13A.4) and (13A.6)

$$J_{N-1} = \min_{\mathbf{u}(k+N-1)} \{\mathbf{x}^T(k+N-1)\mathbf{Q}_{N-1}\mathbf{x}(k+N-1) + \mathbf{u}^T(k+N-1)\Lambda\mathbf{u}(k+N-1)$$
$$+ (\mathbf{A}\mathbf{x}(k+N-1) + \mathbf{B}\mathbf{u}(k+N-1))^T\mathbf{Q}_N(\mathbf{A}\mathbf{x}(k+N-1) + \mathbf{B}\mathbf{u}(k+N-1))\}. \tag{13A.7}$$

The minimization of J_{N-1} with respect to u(k+N-1) implies

$$\frac{\partial J_{N-1}}{\partial u(k+N-1)} = 0.$$

That is

$$u^T(k+N-1)\Lambda + (Ax(k+N-1) + Bu(k+N-1))^T P(k+N)B = 0. \qquad (13A.8)$$

Since Λ and P(k+N) are symmetric, the optimal control input at stage (k+N-1) is given by

$$\begin{aligned} u(k+N-1) &= -\ [\Lambda + B^T P(k+N)B]^{-1} B^T P(k+N)Ax(k+N-1) \\ &= -\ K(k+N-1)x(k+N-1), \end{aligned} \qquad (13A.9)$$

where

$$K(k+N-1) = [\Lambda + B^T P(k+N)B]^{-1} B^T P(k+N)A. \qquad (13A.10)$$

Substituting for u(k+N-1) in (13A.7) from (13A.9) for minimum J_{N-1},

$$\begin{aligned} J_{N-1} = x^T(k+N-1)\ [Q_{N-1} &+ K^T(k+N-1)\Lambda K(k+N-1) \\ &+ (A - BK(k+N-1))^T P(k+N)(A - BK(k+\ N-1))]x(k+N-1). \end{aligned} \qquad (13A.11)$$

Define the term within the bracket as P(k+N-1):

$$\begin{aligned} P(k+N-1) = Q_{N-1} &+ K^T(k+N-1)\Lambda K(k+N-1) \\ &+ (A - BK(k+N-1))^T P(k+N)(A - BK(k+N-1)). \end{aligned}$$

Using (13A.9),

$$\begin{aligned} P(k+N-1) &= Q_{N-1} + K^T(k+N-1)(\Lambda+B^T PB)K + A^T P(k+N)A \\ &\qquad - K^T(k+N-1)B^T P(k+N)A - A^T P(k+N)BK(k+N-1) \\ &= Q_{N-1} + A^T P(k+N)A - A^T P(k+N)B[\Lambda + B^T P(k+N)B]^{-1} B^T P(N)A. \end{aligned} \qquad (13A.12)$$

So, J_{N-1} in (13A.11) can now be expressed as

$$J_{N-1} = x^T(k+N-1)P(k+N-1)x(k+N-1). \qquad (13A.13)$$

Note that equation (13A.13) is the same as (13A.4) except for the time index which is reduced by 1 in (13A.13).

Similarly the total cost over the last three stages (k+N-2, k+N-1, k+N) is given by

$$\begin{aligned} J_{N-2} = \min_{u(k+N-2)} [\{x^T(k+N-2)Q_{N-2}x(k+N-2)+u^T(k+N-2)\Lambda u(k+N-2)\} \\ + \min_{u(k+N-2)} \{J_{N-1}\}]. \end{aligned} \qquad (13A.14)$$

Since

$$x(k+N-1) = Ax(k+N-2) \ + \ Bu(k+N-2),$$

using (13A.13), and substituting for J_{N-1} in (13A.14),

$$J_{N-2} = \min_{u(k+N-2)} \{ x^T(k+N-2) Q_{N-2} x(k+N-2) + u^T(k+N-2) \Lambda u(k+N-2) + (Ax(k+N-2)+Bu(k+N-2))^T P(k+N-1)(Ax(k+N-2)+Bu(k+N-2)) \}. \tag{13A.15}$$

Note that equation (13A.15) is same as (13.7) except for the time index which is reduced by 1 in (13A.15). So, as in (13A.8),

$$\frac{\partial J_{N-2}}{\partial u(k+N-2)} = 0.$$

That is

$$u^T(k+N-2)\Lambda + (Ax(k+N-2) + Bu(k+N-2))^T P(k+N-1)B = 0.$$

Since Λ and $P(k+N-1)$ are symmetric, the optimal control law at stage $(k+N-2)$:

$$\begin{aligned} u(k+N-2) &= - [\Lambda+B^TP(k+N-1)B]^{-1}B^TP(k+N-1)Ax(k+N-2) \\ &= - K(k+N-2)x(k+N-2), \end{aligned} \tag{13A.16}$$

where

$$K(k+N-2) = [\Lambda+B^TP(k+N-1)B]^{-1}B^TP(k+N-1)A. \tag{13A.17}$$

Thus starting from the prespecified terminal condition $P(k+N) = Q_N$, the optimal control law can be computed through backward recursions from one stage to the next, with each stage having identical structure, until the present stage is reached. At each stage k ($i \le k < k+N$, i being the present time), the gain matrix $K(k)$ is computed using $P(k+1)$ and the control $u(k)$ is determined; $P(k)$ is next computed to be used in the next stage $k-1$.

Summing up, the general solution of the deterministic optimal control problem is given by

$$K(k) = [\Lambda + B^TP(k+1)B]^{-1}B^TP(k+1)A, \tag{13A.18}$$

$$P(k) = Q_k + A^TP(k+1)A - A^TP(k+1)B[\Lambda+B^TP(k+1)B]^{-1}B^TP(k+1)A, \tag{13A.19}$$

$$u(k) = - K(k)x(k), \tag{13A.20}$$

where $P(k+N) = Q_N$.

Reference

[1] Bellman, R., and S.E. Dreyfus (1962): *Applied Dynamic Programming*, Princeton University Press, Princeton, N.Y.

APPENDIX 13B

TRANSMITTANCE MATRIX: FORMULATION AND IMPLEMENTATION

13B.1 Introduction

Use of transmittance matrices offers a straightforward method for simplifying a polynomial matrix multiplication problem into an ordinary matrix multiplication problem. The transmittance matrix formulation can be very useful in performing state estimation (Lam, 1982, Clarke *et al*, 1985). This appendix discusses the vector formulation of the transmittance matrix and its application for state estimation; FORTRAN implementation is also presented.

13B.2 Problem Statement

The objective is to compute the transmittance matrix $\mathbf{M}_g$ of $\mathbf{g}(k-1)$ using the following decomposition

$$[I - q^{-1}\mathbf{F}]^{-1}\mathbf{v}g(k-1) \quad = [qI - \mathbf{F}]^{-1}\mathbf{v}g(k) \quad = \mathbf{M}_g\mathbf{g}(k-1)/C(q^{-1}), \qquad (13B.1)$$

where the mxm matrix **F** and the m vector **v** are given by

$$\mathbf{F} = \begin{bmatrix} -c_1 & 1 & 0 & \cdots & 0 \\ -c_2 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \\ -c_n & & & \cdot & \\ 0 & & & \ddots & \\ \vdots & & & & 1 \\ 0 & & 0 & & 0 \end{bmatrix}, \quad \mathbf{v} = [v_1 \ v_2 \ \ldots \ v_m]^T;$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \ldots + c_nq^{-n}, \text{ and}$$

$\mathbf{g}(k-1)$ is the stacked vector of $g(k)$, given by

$$\mathbf{g}(k-1) = [g(k-1) \ g(k-2) \ \ldots \ g(k-m)]^T.$$

Remarks

(a) The transmittance matrix $\mathbf{M}_g$ is a symmetric matrix.

(b) $\mathbf{M}_g$ in (13B.1) is a function of the elements of the vector **v** and the parameters of $C(q^{-1})$.

(c) The actual row (or column) size m of $\mathbf{M}_g$ = max {order of $C(q^{-1})$, the size of **v** without tail-end zeros}. In other words, for implementational purposes, $\mathbf{M}_u$ and $\mathbf{M}_y$ in (13.6.7) need not be of the same size.

13B.3 Solution of $[qI - F]^{-1}$

$$[qI - F]^{-1} = \text{Adjoint } [qI - F]/D(q), \tag{13B.2}$$

where mxm matrix

$$[qI - F] = \begin{bmatrix} q+c_1 & -1 & & & \\ c_2 & q & -1 & 0 & \\ c_n & 0 & \cdot & & \\ \vdots & \vdots & & \cdot & \\ 0 & 0 & & \cdot & -1 \\ 0 & 0 & & & q \end{bmatrix}, \tag{13B.3}$$

and D(q), the determinant of [qI - F], is given by

$$\begin{aligned} D(q) &= q^m + c_1 q^{m-1} + \ldots + c_n q^{m-n} \\ &= q^m C(q^{-1}). \end{aligned} \tag{13B.4}$$

It is known (Cadzow and Martens, 1970) that

$$\begin{aligned} \text{Adjoint } [qI - F] &= Iq^{m-1} + H_1 q^{m-2} + \ldots + H_{m-1} \\ &= q^m[Iq^{-1} + H_1 q^{-2} + H_2 q^{-3} + \ldots + H_{m-1}], \end{aligned} \tag{13B.5}$$

where

$$\begin{aligned} H_1 &= F + p_1 I, & p_1 &= \text{trace}(F), \\ &= F + c_1 I. & & \\ H_2 &= FH_1 + p_2 I. & p_2 &= 1/2 \text{ trace}(FH_1) = c_2. \\ H_3 &= FH_2 + p_3 I. & p_3 &= 1/3 \text{ trace}(FH_2) = c_3. \\ &\vdots & &\vdots \\ H_m &= FH_{m-1} + p_m I. & p_m &= 1/m \text{ trace}(FH_m) = c_m. \end{aligned}$$

In the present case $p_{n+1} = p_{n+2} = \ldots = p_m = 0$, where n = degree of $C(q^{-1})$. Following (13B.2 - 13B.5),

$$\begin{aligned} [qI - F]^{-1} &= [Iq^{-1} + H_1 q^{-2} + \ldots + H_{m-1}]/C(q^{-1}) \\ &= M/C(\bar{q}^{1}\ ), \end{aligned} \tag{13B.6}$$

where

$$M = \begin{bmatrix} q^{-1} & q^{-2} & q^{-3} & & \cdot & \cdot & \cdot & q^{-m} \\ s_2 & q^{-1}r_2 & q^{-2}r_2 & & \cdot & \cdot & \cdot & q^{-m+1}r_2 \\ qs_3 & s_3 & q^{-1}r_3 & & \cdot & \cdot & \cdot & q^{-m+2}r_3 \\ \vdots & \ddots & \ddots & \ddots & & & & \vdots \\ q^{n-2}s_n & q^{n-3}s_n \cdot & s_n & q^{-1}r_n & \cdot & \cdot & \cdot & q^{-m+n-1}r_n \\ & & & & q^{-1}r_{n+1} & & \cdot & q^{-m+n}r_{n+1} \\ & 0 & & & & \cdot & & q^{-m+n+1}r_{n+1} \\ & & & & & & \cdot & \vdots \\ & & & & & & & q^{-1}r_{n+1} \end{bmatrix}, \tag{13B.7a}$$

$$
\begin{aligned}
r_i &= 1 + c_1q^{-1} + \dots + c_{i-1}q^{-i+1}, \text{ for } i = 2,3,.., n+1,\\
r_{n+2} &= r_{n+3} = \dots = r_{n+1},\\
s_i &= r_i - C(q^{-1}),\\
s_{n+1} &= s_{n+2} = \dots = 0. && (13B.7b)
\end{aligned}
$$

The derivation of the transmittance matrix from **M** can be illustrated as follows.

Example 13B.3 The state-space representation of the process

$$y(k) + a_1y(k-1) + a_2y(k-2) = b_0u(k-1) + b_1u(k-2) + b_2u(k-3)$$
$$+ e(k) + c_1e(k-1) + c_2e(k-2),$$

is given by

$$
\begin{aligned}
\mathbf{x}(k+1) &= \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} b_o \\ b_1 \\ b_2 \end{bmatrix}u(k) + \begin{bmatrix} c_1-a_1 \\ c_2-a_2 \\ 0 \end{bmatrix}e(k),\\
y(k) &= [1,\ 0\ \dots\ 0]^T\mathbf{x}(k) + e(k);
\end{aligned}
\qquad (13B.8)
$$

find $\mathbf{M}_u$, where

$$[q\mathbf{I} - \mathbf{F}]^{-1}\mathbf{b}u(k) = \mathbf{M}_u u(k-1)/C(q^{-1}), \text{ and} \qquad \mathbf{F} = [\mathbf{A} - \mathbf{e}\mathbf{c}^T].$$

Here n = 2, k = 1, m = 3.

$$[q\mathbf{I} - \mathbf{F}] = \begin{bmatrix} q+c_1 & -1 & 0 \\ c_2 & q & -1 \\ 0 & 0 & q \end{bmatrix} \qquad (13B.9)$$

$$
\begin{aligned}
D(q) &= q^3(1 + c_1q^{-1} + c_2q^{-2})\\
[q\mathbf{I} - \mathbf{F}]^{-1} &= [\mathbf{I}q^{-1} + \mathbf{H}_1q^{-2} + \mathbf{H}_2q^{-3}]/C(q^{-1}).\\
\mathbf{H}_1 &= \mathbf{F} + p_1\mathbf{I}, \qquad p_1 = \text{trace}(\mathbf{F}) = c_1.\\
\mathbf{H}_2 &= \mathbf{F} + p_2\mathbf{I}, \qquad p_2 = 1/2\ \text{trace}(\mathbf{F}\mathbf{H}_1) = c_2.
\end{aligned}
$$

That is

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 1 & 0 \\ -c_2 & c_1 & 1 \\ 0 & 0 & c_1 \end{bmatrix}, \qquad \mathbf{H}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & c_1 \\ 0 & 0 & c_2 \end{bmatrix}.$$

Following (13B.6), $[q\mathbf{I} - \mathbf{F}]^{-1} = \mathbf{M}/C(q^{-1})$, where

$$\mathbf{M} = \begin{bmatrix} q^{-1} & q^{-2} & q^{-3} \\ -c_2q^{-2} & q^{-1}+c_1q^{-2} & q^{-2}+c_1q^{-3} \\ 0 & 0 & q^{-1}+c_1q^{-2}+c_2q^{-3} \end{bmatrix}. \qquad (13B.10)$$

Alternatively, using (13B.7) directly,

$$r_2 = 1 + c_1q^{-1}, \qquad r_3 = 1 + c_1q^{-1} + c_2q^{-2},$$

$$s_2 = -c_2q^{-1}, \qquad s_3 = 0,$$

$$\mathbf{M} = \begin{bmatrix} q^{-1} & q^{-2} & q^{-3} \\ s_2 & q^{-1}r_2 & q^{-2}r_2 \\ qs_3 & s_3 & q^{-1}r_3 \end{bmatrix},$$

which is the same as (13B.10).

Now let us consider a general expression for

$$[q\mathbf{I} - \mathbf{F}]^{-1}\mathbf{v}g(k), \qquad \text{where} \qquad \mathbf{v} = [v_1 \ v_2 \ v_3]^T.$$

Using (13B.10),

$$[q\mathbf{I}-\mathbf{F}]^{-1}\mathbf{v}g(k)=\frac{1}{C(q^{-1})}\begin{bmatrix} q^{-1} & q^{-2} & q^{-3} \\ -c_2q^{-2} & q^{-1}+c_1q^{-2} & q^{-2}+c_1q^{-3} \\ 0 & 0 & q^{-1}+c_1q^{-2}+c_2q^{-3} \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}g(k)$$

$$=\frac{1}{C(q^{-1})}\begin{bmatrix} v_1q^{-1} + v_2q^{-2} + v_3q^{-3} \\ v_2q^{-1} + (c_1v_2-c_2v_1+v_3)q^{-2} + c_1v_3q^{-3} \\ v_3\, q^{-1} + c_1\, v_3q^{-2} + c_2v_3q^{-3} \end{bmatrix}g(k)$$

$$=\frac{1}{C(q^{-1})}\begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & v_3+c_2v_1-c_2v_2 & c_1v_3 \\ v_3 & c_1v_3 & c_2v_3 \end{bmatrix}\begin{bmatrix} v(k-1) \\ v(k-2) \\ v(k-3) \end{bmatrix}, \qquad (13B.11)$$

Thus

$$[q\mathbf{I} - \mathbf{F}]^{-1}\mathbf{v}g(k) = \mathbf{M}_g\mathbf{g}(k-1)/C(q^{-1}), \qquad (13B.12)$$

where

$$\mathbf{g}(k-1) = [g(k-1) \quad g(k-2) \quad g(k-3)]^T,$$

and 3x3 matrix $\mathbf{M}_g$, defined as the transmittance matrix, is given by (13B.11).

For the present case (13B.8), since

$$\mathbf{b} = [b_o \quad b_1 \quad b_2]^T \quad \text{and} \quad [q\mathbf{I} - \mathbf{F}]^{-1}\mathbf{b}u(k) = \mathbf{M}_u\mathbf{u}(k-1)/C(q^{-1}),$$

the transmittance matrix $\mathbf{M}_u$ is given by

$$\mathbf{M}_u = \begin{bmatrix} b_0 & b_1 & b_2 \\ b_1 & b_2+c_1b_1-c_2b_0 & c_1b_2 \\ b_2 & c_1b_2 & c_2b_2 \end{bmatrix},$$

with

$$u(k-1) = [u(k-1) \quad u(k-2) \quad u(k-3)]^T.$$

13B.4 Structure of the Transmittance Matrix

As shown in the last example, the transformation

$$[qI - F]^{-1}vg(k) = [I - q^{-1}F]^{-1}vg(k-1) = M_g g(k-1)/C(q^{-1})$$

converts a polynomial matrix multiplication problem into an ordinary matrix multiplication problem; $1/C(q^{-1})$ is handled separately. The transmittance matrix M_g is of the same size as F. The transmittance matrix is symmetric and has a general structure as follows

$$M_g = \begin{bmatrix} v_1 & v_2 & v_3 & \cdots & v_{m-1} & v_m \\ * & \begin{array}{c} v_3+c_1v_2 \\ -c_2v_1 \end{array} & \begin{array}{c} v_4+c_1v_3 \\ -c_3v_1 \end{array} & \cdots & \begin{array}{c} v_m+c_1v_{m-1} \\ -c_{m-1}v_1 \end{array} & \begin{array}{c} c_1v_m \\ -c_mv_1 \end{array} \\ * & * & \begin{array}{c} v_5+c_1v_4+c_2v_3 \\ -c_4v_1-c_3v_2 \end{array} & \cdots & \begin{array}{c} c_1v_m+c_2v_{m-1} \\ -c_mv_1-c_{m-1}v_2 \end{array} & \begin{array}{c} c_2v_m \\ -c_mv_2 \end{array} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & * & \cdots & \begin{array}{c} c_{m-3}v_m-c_{m-2}v_{m-1} \\ -c_mv_{m-3}-c_{m-1}v_{m-2} \end{array} & \begin{array}{c} c_{m-2}v_m \\ -c_mv_{m-2} \end{array} \\ * & * & * & \cdots & * & \begin{array}{c} c_{m-1}v_m \\ -c_mv_{m-1} \end{array} \end{bmatrix}. \qquad (13B.13)$$

13B.5 Implementation

In the transformation

$$[qI - F\ ]^{-1}vg(k) = M_g g(k-1)/C(q^{-1}),$$

the transmittance matrix M_g is given by (13.B13), the parameter vector **v** and the stacked vector **g**(k-1) are given by

$$\mathbf{v} = [v_1 \ v_2 \ \dots \ v_m]^T,$$

$$\mathbf{g}(k-1) = [g(k-1) \ g(k-2) \ \dots \ g(k-m)]^T.$$

The transmittance matrix being symmetric, the diagonal and the upper triangular parts are stored in a vector TM() of size m(m+1)/2 as shown below.

$$M_g(.) = \begin{bmatrix} TM(1) & TM(2) & TM(4) & TM(7) & TM(11) \\ * & TM(3) & TM(5) & TM(8) & TM(12) \\ * & * & TM(6) & TM(9) & TM(13) \\ * & * & * & TM(10) & TM(14) \\ * & * & * & * & TM(15) \end{bmatrix} \longrightarrow TM\ ().$$

13B.6 FORTRAN Mechanization

The FORTRAN mechanization has two parts:
(i) formation of the transmittance matrix, and
(ii) multiplication of the transmittance matrix by a vector.

Inputs V() parameter vector v
C(),IC C vector of size IC containing parameters $c_1, c_2, \dots$ of $C(q^{-1})$
ITM Size of transmittance matrix vector TM(), = (>1=) max{degree of $C(q^{-1})$, size of V()}
G() Stacked vector of process variable, post-multiplying TM, e.g., $\Delta u(k-1)$ in (13.6.34).
PXNEW() Zero vector initially;
after 1st pass, = TM()*G() and so on.
Outputs TM() Linear array transmittance matrix elements
PXNEW() updated by additional set of TM()*G().

Comment: TM() stores the diagonal and upper triangular part of the transmittance matrix.

```
      IC1          = IC + 1
      N            = ITM +1
      MTOTAL       = (ITM*N)/2
      DO 2   I     = 3, MTOTAL
2     TM(I)        = 0.0
      K            = (ITM+1)/2
      L            = 0
      DO   4 I     = 1,K
      L            = L + I
      IW           = L
      DO 4 J       = I, N - I
      TM(IW)       = V(I+J-1)
4     IW           = IW + J
```

Comment: Now C_i*v_i elements are placed. Start from the last column of the 2nd row and proceed m-1 steps down the column; next start from last but one column of the 2nd row and proceed m-2 steps down the column etc.

```
      ITER         = ITM/2
      M            = MTOTAL + N
      DO 6 L       = 1, ITER
      M            = M + L - N
      ILIM         = N - L - L
      K            = M
      DO 6 I     = 1, ILIM
      K          = K - ITM + I + L - 2
      NL         = N - I
      DO 6 J     = 1, ILIM + 1 - I
      IW         = K + J
      IF (J.LT.IC1) TM(IW) = TM(IW) + C(J) * V(NL)
      IF (NL.LT.IC1) TM(IW) = TM(IW) - C(NL) * V(J)
6     CONTINUE
```

Comment: Now compute PXNEW() = (transmittance matrix) x (stacked vector) + earlier PXNEW(). For example consider computation of (13.6.12):

$$C(q^{-1})\hat{x}(k|k-1) = [M_u\Delta u(k-1) + M_w\Delta w(k) + M_e\underline{\varepsilon}(k-1)];$$

(i) first PXNEW(I) is zeroed, (ii) call present subroutine to compute $M_u\Delta u(k-1)$ which is added to PXNEW(), and (iii) repeat (ii) for $M_w\Delta w(k)$ and for $M_e\underline{\varepsilon}(k-1)$. M_u, M_w and M_e need not be of the same size.

```
      L          = 0
      DO 9 I     = 1, ITM
      DO 92 J    = 1, I
92    PXNEW(I)   = PXNEW(I) + TM(L + J)*G(J)
      IW         = L + I
      DO 94 K    = 1, ITM - I
      IW         = IW + I + K - 1
94    PXNEW(I)   = PXNEW(I) + TM(IW) *G(I+K)
      L = L + I
9     CONTINUE
```

References

[1] Cadzow, J.A., and H.R. Martens (1970): *Discrete-time and Computer Control Systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J.

[2] Clarke, D.W., P.P. Kanjilal, and C. Mohtadi, (1985): 'A Generalized LQG Approach to Self-tuning Control, Part II. Implementation and Simulation', *International Journal of Control*, 41, 1525-1544.

[3] Lam, K. P. (1982): 'Design of stochastic discrete time linear optimal regulators, Part I', *International Journal of Systems Science*, 13, 979-1000.

APPENDIX 13C

COVARIANCE TIME UPDATE USING U-D FACTORIZATION

13C.1 Introduction

U-D covariance factorization is attractive because of numerical robustness, algorithmic simplicity and computational efficiency. Bierman (1977, p.124) studies the general covariance time update problem:

$$\mathbf{P} = \phi \mathbf{P}^* \phi^T + \mathbf{GQG}^T, \qquad (13C.1)$$

where U-D factors of $\mathbf{P}^*$, $\mathbf{P}^* = \mathbf{U}^*\mathbf{D}^*\mathbf{U}^{*T}$, are given and the updated U-D factors are computed. A particular form of the U-D covariance update problem relating to the LQ state-space controller (Sec.13.7) is discussed in Clarke *et al* (1985), where the vector implementation is also presented; the material presented here is largely based on this reference.

13C.2 The Problem

$$\begin{aligned} \mathbf{P} &= \mathbf{A}^T\mathbf{P}^*\mathbf{A} + \mathbf{g}q\mathbf{g}^T, & \mathbf{P}^* = \mathbf{U}^*\mathbf{D}^*\mathbf{U}^{*T} \quad (13C.2a)\\ &= \mathbf{UDU}^T, & (13C.2b) \end{aligned}$$

where $\mathbf{A}$ is in observable canonical form, and $\mathbf{g} = [1\ 0\ldots0]^T$; the objectives are
(a) to compute updated factors, $\mathbf{U}$ and $\mathbf{D}$, given prior covariance factors, $\mathbf{U}^*$ and $\mathbf{D}^*$, where $\mathbf{D}$ and $\mathbf{D}^*$ are diagonal and $\mathbf{U}$ and $\mathbf{U}^*$ are unit upper triangular matrices, and
(b) to obtain the vector implementation for the covariance U-D time update (13C.2).

13C.3 Summary of Algorithm

The factorization problem may be reformulated as follows. The updated nxn covariance matrix $\mathbf{P}$ is given by

$$\begin{aligned} \mathrm{P}(k) &= [\mathbf{g}\quad \mathbf{A}^T\mathbf{U}^*]\ \mathrm{diag}\{q\quad \mathbf{D}^*\}\ [\mathbf{g}\quad \mathbf{A}^T\mathbf{U}^*]^T,\\ &= \mathbf{W}\bar{\mathbf{D}}\mathbf{W}^T, \text{ i.e.,}\qquad \mathbf{W} = [\mathbf{g}\quad \mathbf{A}^T\mathbf{U}],\ \bar{\mathbf{D}} = \mathrm{diag}[q\ \mathbf{D}^*], \qquad (13C.3)\\ &= \mathbf{UDU}^T, \qquad (13C.4) \end{aligned}$$

where the square matrix $\bar{\mathbf{D}}$ is m diagonal, $\mathbf{D}$ is n diagonal and $\mathbf{W}$ is an nxm matrix, m>n.

One of the efficient ways of U-D factorization in the present case is by using modified weighted Gram-Schmidt (mWGS) orthogonalization procedure. The basic idea is to decompose **W** in (13C.3) such that

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_n^T \end{bmatrix} = \mathbf{U} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_n^T \end{bmatrix}, \qquad (13C.5)$$

where $\mathbf{v}_i$ are weighted orthogonal m vectors. The elements (D_j) of the diagonal matrix **D** are computed from

$$\mathbf{v}_i^T\bar{\mathbf{D}}\mathbf{v}_j = D_j\delta_{ij}, \qquad \delta_{ij} = 0, \text{ if } i \neq j, \\ = 1, \text{ if } i = j. \qquad (13C.6)$$

Backward iteration is used to compute the **U** and **D** factors. First $\mathbf{v}_n$ is defined as $\mathbf{w}_n$ and then from each lower vector $\mathbf{v}_i$ $(=\mathbf{w}_i)$, $(i = n-1,\ldots, 1)$, the part that is orthogonal to $\mathbf{v}_n$ is extracted and this process continues individually through all the lower vectors $\mathbf{v}_{n-1},\ldots, \mathbf{v}_2$.
For $j = n, n-1,\ldots, 1$, initialize $\mathbf{v}_j^i = \mathbf{w}_j$, $i = n$, where the superscript i represents (n+1-i)th stage of the computational process. For $j = n, n-1,\ldots, 2$, iterate through the following steps.

$$D_j = (\mathbf{v}_j^j)^T\bar{\mathbf{D}}\mathbf{v}_j^1,$$

$$U_{kj} = (\mathbf{v}_k^j)^T\bar{\mathbf{D}}\mathbf{v}_j^j/D_j, \qquad k = 1,\ldots, j-1,$$

and carry over

$$\mathbf{v}_k^{j-1} = \mathbf{v}_k^j - U_{kj}\mathbf{v}_j^j, \qquad k = 1,\ldots, j-1.$$

Finally, $D_1 = (\mathbf{v}_1^1)^T\bar{\mathbf{D}}\mathbf{v}_1^1$.

13C.4 Implementation

The FORTRAN implementation in matrix formulation for the general case (13C.1) is given in Bierman (1977, p.131). The vector implementation is presented here for the case when **A** is in the observable canonical form and $\mathbf{g} = [1\ 0\ldots\ 0]^T$.

Given $\mathbf{U}^*$ and $\mathbf{D}^*$ factors of $\mathbf{P}^*$: $\mathbf{P}^* = \mathbf{U}^*\mathbf{D}^*\mathbf{U}^{*T}$, the problem is to compute the updated **U** and **D** factors in

$$\mathbf{P} = \mathbf{A}^T\mathbf{P}^*\mathbf{A} + \mathbf{g}q\mathbf{g}^T = [\mathbf{g}\quad \mathbf{A}^T\mathbf{U}^*]\ \mathrm{diag}\{q\quad \mathbf{D}^*\}\ [\mathbf{g}\ \mathbf{A}^T\mathbf{U}^*]^T, \\ = \mathbf{W}\bar{\mathbf{D}}\mathbf{W}^T = \mathbf{U}\mathbf{D}\mathbf{U}^T.$$

The implementation has two main parts:
(i) formation of **W** as $\mathbf{A}^T\mathbf{U}^*$ and $\bar{\mathbf{D}} = \mathrm{diag}\ \{q\quad \mathbf{D}^*\}$,
(ii) extraction of **U** and **D** factors using mWGS algorithm.

As discussed in Sec.13.7.2 and shown in (13.6.40)-(13.6.41), **W** has a strictly zero lower triangular part with unity upper diagonal elements. The upper triangular elements of **W** are stored in the vector **w**; for example for a 3x3 **P** matrix, **W** and $\bar{\mathbf{D}}$ elements are stored in **w** and $\bar{\mathbf{d}}$ vectors typically as follows:

$$\mathbf{W} = \begin{bmatrix} 1 & w(1) & w(2) & w(3) \\ 0 & 1 & w(3) & w(5) \\ 0 & 0 & 1 & w(6) \end{bmatrix}, \qquad \bar{\mathbf{d}} = [q \; d_1^* \; d_2^* \; d_3^*]. \qquad (13C.7)$$

For an nxn **P**, the necessary size of **w** is n(n+1)/2.

13C.5 FORTRAN Mechanization

Inputs A() unsigned first column of **A**, for example in (13.6.40): A() = $(a_1 \; a_2 \; a_3)$

N size of the covariance matrix

U(),D() **U*** and **D*** factors of **P*** to be time updated, stored as vectors

q as stated in the update equation (13C.2), default=1.

Outputs U(),D() time updated U-D factors of **P**.

Comment: Upper triangular elements of **W** are stored in W(); INDEX() is a vector of integer values needed to find the index for the elements of **W** as in (13C.7).

```
      N1          = N+1
      N2          = N+2
      NTOTAL      = (N*N1)/2
      INDEX(1)    = 0
      DO 3 I      = 2, N
3     INDEX(I)    = N1 - I + INDEX (I-1)
```

Comment: Now store the upper triangular part of **W** in W() and Diag. {q,D} in D().

```
      W(1)        = -A(1)
      D(N1)       = D(N)
      KU          = 0
      DO 4 J      = 2,N
      SI          = 0.0
      DO 45 I     = 1, J-1
      KU          = KU + 1
      W(J+KU)     = U(KU)
45    SI          = SI-U(KU)*A(I)
      W(KU+1)     = SI - A(J)
      D(N2-J)     = D(N1-J)
4     CONTINUE
```

```
     D(1)        = q
```

Comment: Up to (13C.3) is completed here. Now compute the updated **U** and **D** factors and store in U() and D().

```
1000 DO 6 ITER  = 1, N
     J          = N-ITER + 1
     NJ         = N1 - J
     SI         = D(J)
     DO 62 K    = 1, NJ
     IW         = NTOTAL - INDEX(K)
     V(K)       = W(IW)
     A(K)       = D(N2 - K) * V(K)
62   SI         = SI + V(K) * A(K)
```

Comment: K indexing starts from the rightmost side of **W**.

```
     A(NJ+1)        =  D(J)
     V(NJ+1)         = 1.0
     DNEW(J)         = SI
     IF(J.EQ.1) GO TO 6
     IF(SI.LT.1E-15) GO TO 6
     DIV       = 1.0 / SI
     NJ        = NJ + 1
     JM        = J - 1
     DO 64 K   = 1, JM
     SI        = 0.0
     MLAST     = NTOTAL - K
     DO 645 I  = 1, NJ
     IW        = MLAST - INDEX(I)
645  SI        = SI + W(IW)*A(I)
     SI        = SI*DIV
     U(IW)     = SI
     DO 64 I   =  1, NJ
     IW         = MLAST - INDEX(I)
64   W(IW)      = W(IW) - SI*V(I)
6    NTOTAL     = NTOTAL - 1
     DO 8 I     = 1, N
     D(I)       = DNEW(I)
8    CONTINUE
```

References

[1] Bierman, G.J. (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[2] Clarke, D.W., P.P. Kanjilal, and C. Mohtadi (1985): 'A generalized LQG approach to self-tuning control, Part II. Implementation and simulation', *Int. J. Control*, 41, 1525–1544.

Appendix 14A

LOW-PASS FILTER

A low-pass filter is a frequency domain filter, which allows the specified low-frequency part of the signal or data sequence to pass through, whereas the higher frequency components are attenuated. The objective is to separate from the data undesirable high frequency components, which may be due to external disturbances or noise. The filter may be characterized by the pass-band, the transition band, the stop-band and the gain or the pass-band magnitude (see Fig.14A.1). The smaller the transition region, the sharper is the separation between the frequency components passed and those attenuated.

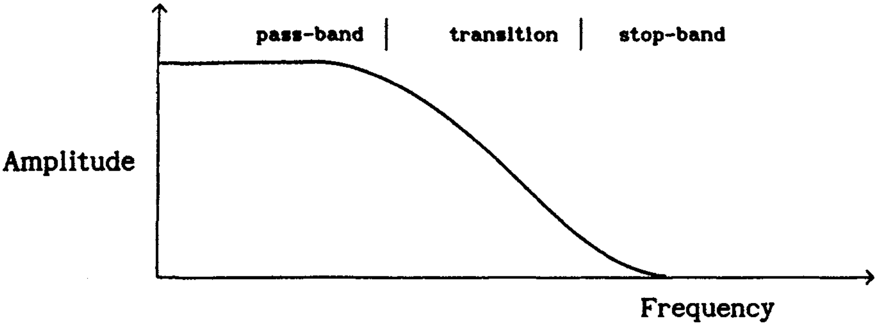
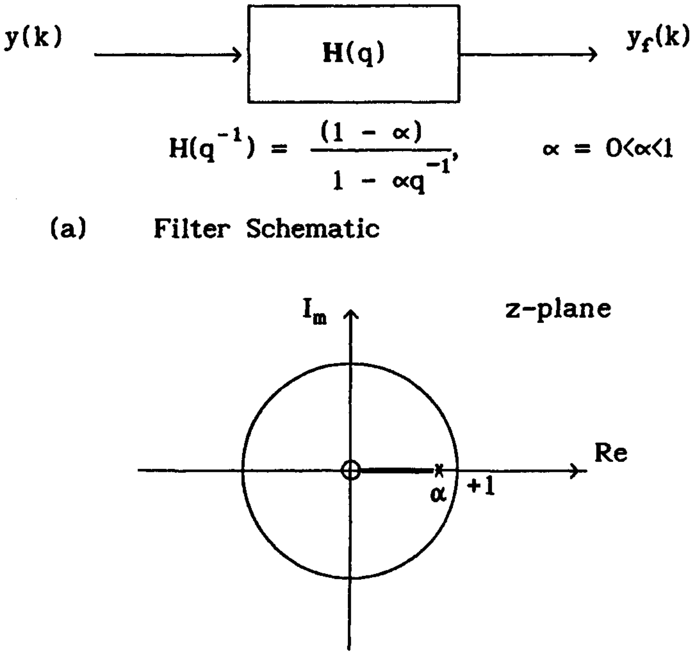


Figure 14A.1 Typical frequency response of a low-pass filter.

If T is the sampling period, that is the time-interval at which data are received, the maximum frequency component in the data = $\omega_s/T = 2\pi/2T$, ω_s being the sampling frequency. The cut-off frequency, ω_c, has to be lower than $\omega_s/2$. For example, for a yearly periodic process with monthly data, T = 1/12; the highest frequency component will be the bimonthly periodic component. So if frequency components are of interest, the low-pass filter may be used with a pass-band of 0 - $\omega_s/4$.

There are two basic types of digital (or discrete-time) filters: nonrecursive and recursive. A *nonrecursive filter* generates the output from the inputs or time-delayed inputs.



$$H(q^{-1}) = \frac{(1-\alpha)}{1-\alpha q^{-1}}, \qquad \alpha = 0<\alpha<1$$

(a) Filter Schematic

(b) Pole-zero diagram in z-plane

Figure 14A.2 A first order low-pass filter.

For example,

$$y_f(k) = b_1 y(k) + b_2\ y(k-1) + \ldots + b_n y(k-n). \qquad (14A.1)$$

A *recursive filter* generates the output from the present and the time-delayed inputs as well as the time-delayed outputs. For example,

$$y_f(k) = b_1 y(k) + b_2 y(k-1) + \ldots\ldots + b_n y(k-n)$$
$$- a_1 y_f(k-1) - a_2 y_f(k-2) - \ldots - a_n y_f(k-n). \qquad (14A.2)$$

A nonrecursive filter is also called finite impulse response (FIR) filter, whereas the recursive filter is also referred to as infinite impulse response (IIR) filter. The impulse response for a recursive filter takes a long time to die because of the AR part of the filter equation in (14A.2).

Remark
A moving average system is a FIR system, whereas a system with autoregressive part (e.g. AR, ARMA systems) is an IIR system.

□□

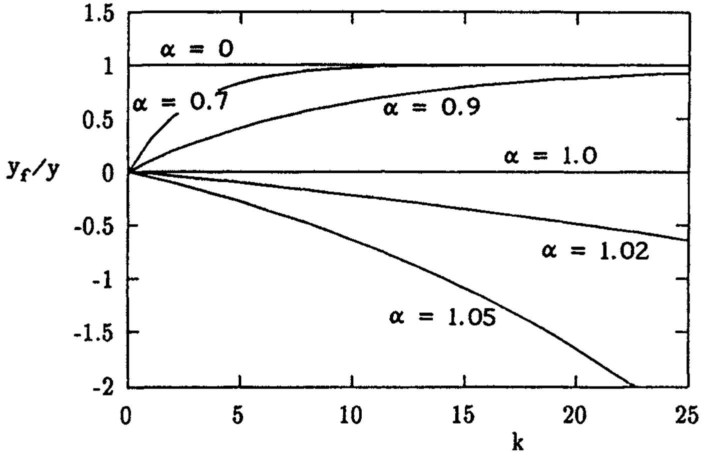


Figure 14A.3 The first-order low-pass filter characteristic for different values of α.

A first order recursive low-pass filter, H(q) (Fig. 14A.2) can be expressed as

$$y_f(k) = H(q)y(k)$$

$$= \frac{(1-\alpha)}{1-\alpha q^{-1}}\, y(k) = \frac{(1-\alpha)q}{q-\alpha}\, y(k), \qquad 0<\alpha<1, \tag{14A.3}$$

where q is the unit forward shift operator qy(k) = y(k+1).

The roots of the denominator and of the numerator of H(q) are called poles and zeros respectively. The filter-characteristic with respect to different values of α is shown in Fig.14A.3.

As shown in the figure, the response will be unstable for $\alpha > 1$. For $\alpha < 1$, lower values of α will lead to the pole being further away on the real axis from +1 point of the unit circle in the z-plane (see Fig.14A.2); such a pole is known as a *fast pole*, because it leads to faster response. The closer the pole is to +1 point, the slower it will be, resulting in a relatively sluggish response.

The performance of the low-pass filter is judged by the ripples in the pass-band and the sharpness of cut-off in the frequency response of the filter. Two standard types of low-pass filters are (a) the Butterworth filter and (b) the Chebyshev filter. The Butterworth filter produces a maximally flat response in the pass-band but a relatively large transition band. The Chebyshev filter has a smaller transition band than the same order Butterworth filter (i.e. it has sharper cutoff) but it has ripples either in stop-band or pass-band.

The design procedures of the low-pass filters are detailed in the following references.

References

[1] Oppenheim, A.V., and R.W. Schàfer (1975): *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J.

[2] Taylor, F.J. (1983): *Digital Filter Design Handbook*, Marcel Dekker, New York.

[3] Williams, C.S. (1986): *Designing Digital Filters*, Prentice-Hall, Englewood Cliffs, N.J.

APPENDIX 14B

PERMEABILITY DATA

The following sets of data are the 2-minutely recorded permeability measurements of the green-mix permeability in the process of iron-ore sintering collected from an iron and steel plant.

Here the data for ten consecutive hours are presented column wise; each column contains data (expressed in permeability index or P.I.) for one hour.

Table 14B.1 Green-mix permeability (P.I.)

29.5	29.5	32.0	26.9	23.4	27.3	23.5	24.0	26.0	27.0
29.4	31.9	27.0	28.0	26.4	28.3	28.0	26.0	24.2	33.0
32.1	29.9	28.1	28.5	23.5	25.0	21.5	22.5	22.5	23.6
29.0	30.5	31.4	28.8	24.2	24.2	24.0	23.0	24.5	23.3
31.1	31.0	29.0	27.2	24.3	25.0	27.0	22.5	26.5	27.5
30.5	31.5	31.5	29.0	25.5	24.0	26.8	23.2	22.0	25.0
30.0	33.5	29.2	30.2	25.6	23.5	28.3	25.4	22.3	21.0
34.5	28.4	33.8	31.4	24.0	23.0	25.0	21.3	23.0	25.5
30.2	31.4	27.3	28.4	21.2	23.2	24.0	20.2	25.6	26.3
30.3	31.0	31.0	30.0	28.0	25.3	25.3	22.0	27.4	25.0
32.7	31.9	27.0	30.2	23.5	24.0	25.2	23.0	25.0	23.5
33.0	31.0	29.0	31.4	26.5	27.5	21.7	22.9	24.3	25.0
33.6	29.0	31.0	30.0	26.3	21.8	23.8	25.3	23.7	26.2
31.0	32.6	26.5	29.0	29.5	23.0	23.2	25.0	26.2	25.0
30.5	30.5	27.4	28.0	23.0	24.3	26.2	26.0	28.0	22.8
30.6	31.5	28.2	25.5	31.5	26.3	22.3	24.0	24.0	20.8
27.8	31.0	29.0	27.0	23.5	24.0	28.0	22.0	23.5	22.9
37.0	34.5	28.6	25.0	27.5	23.0	20.8	21.0	26.6	28.1
29.3	30.0	28.4	28.0	26.0	20.8	26.0	22.0	25.8	25.7
34.8	32.0	28.6	24.3	26.5	24.0	26.5	24.2	24.0	23.1
32.0	33.8	27.0	26.3	25.5	25.0	26.2	20.5	20.7	26.0
34.0	31.5	29.3	24.0	26.5	26.2	23.5	24.5	30.5	27.2
33.0	34.5	28.5	29.0	25.3	27.3	27.5	25.0	25.5	22.8
34.5	31.0	30.8	32.5	25.2	26.9	23.0	24.5	25.8	21.2
32.5	33.0	27.0	27.0	25.0	34.8	23.8	25.0	23.0	28.8
34.5	30.7	24.0	25.5	24.3	21.5	24.0	27.3	28.3	23.7
31.2	34.3	25.5	26.0	24.2	25.3	21.5	24.5	33.3	26.8
32.5	31.0	28.0	29.0	29.3	26.3	22.4	22.2	28.0	24.0
31.7	30.5	29.2	32.5	28.0	25.0	21.8	26.0	27.2	22.5
33.0	31.5	28.7	27.0	26.0	23.5	22.5	24.2	26.5	23.0

APPENDIX 14C

COMPOSITE DATA ON MATERNAL ECG CONTAINING FETAL ECG

This set of data were recorded from the abdominal lead of an expectant mother during the 37th week of the gestation period. These data were recorded with an amplifier gain of 10,000 and 3dB bandwidth of 0.05 to 250 Hz. The data were digitized at a sampling rate of 500 Hz. The data presented here are obtained by downsampling the digitized data by a factor of 4.

The data are presented serially columnwise, separately on each page.

Table 14C.1 Data on composite maternal ECG

15	18	8	13	8	8	11	2	-2	-5
-12	-5	-3	-1	0	3	18	1	12	23
21	24	28	35	18	22	18	13	15	9
8	24	6	12	3	3	10	-2	8	10
10	29	-67	-46	-22	-4	-1	-13	-13	-16
-19	-13	-17	-10	-16	-22	-14	-27	-14	-14
-29	-17	-18	-22	-41	-56	-36	-1	82	118
76	25	-6	-10	-11	-20	-1	-20	-16	-15
-15	4	-6	-12	8	-3	4	3	4	6
3	19	-14	-71	-32	-14	9	2	-2	3
-2	14	5	13	24	10	18	16	19	19
12	29	23	25	27	29	41	30	34	37
34	34	39	39	48	44	40	58	31	38
41	47	57	51	56	47	54	54	50	53
42	72	37	-27	-2	-3	22	49	106	172
152	74	38	25	20	14	11	11	8	9
18	18	18	11	20	7	10	10	18	18
18	10	28	10	14	5	15	16	0	12
4	4	5	1	3	6	15	11	16	17
5	26	19	-57	-32	-8	8	4	1	14
-5	8	-1	-10	-11	-11	-12	-7	-17	-14
-18	-11	-7	-8	1	-4	-1	0	2	10
-1	0	5	2	-8	-26	-26	1	44	134
134	83	32	10	14	-2	6	-5	2	10
6	17	32	-18	-44	-6	16	18	15	15
2	18	17	10	15	14	22	25	8	26
18	11	11	15	26	25	28	19	24	28
23	26	29	16	20	18	23	21	6	13
12	12	19	23	32	26	27	28	18	28
24	41	47	-2	-26	11	40	48	33	42
41	27	32	31	38	33	19	12	20	63
139	173	163	89	42	40	20	39	31	34
42	45	45	40	38	42	42	47	32	38
32	38	40	35	51	48	48	54	44	56
49	66	56	-19	10	43	58	59	58	52
56	47	48	43	51	39	37	38	29	35
36	33	40	42	41	31	25	35	25	21
20	11	26	20	27	19	18	15	2	11
12	4	8	8	11	12	12	4	-26	-7
21	101	133	102	72	47	15	14	7	11

(Contd.)

Table 14C.1 Data on composite maternal ECG (contd.)

9	-2	9	-3	-6	2	-18	-4	-8	-10
-8	-21	-13	-12	-13	-6	-18	-10	-18	-8
-11	-26	-21	-28	-28	-18	-21	-11	-26	-15
-11	-23	-9	-18	-13	-18	-32	-24	-26	-22
-21	-88	-90	-54	-36	-36	-48	-46	-60	-56
-58	-58	-59	-54	-51	-56	-54	-50	-55	-56
-52	-50	-53	-50	-53	-49	-53	-64	-76	-81
-45	22	118	104	24	-20	-48	-53	-53	-53
-44	-58	-50	-48	-47	-43	-48	-44	-28	-37
-98	-102	-62	-47	-50	-43	-53	-39	-46	-38
-42	-46	-48	-48	-58	-47	-48	-34	-45	-45
-38	-48	-46	-48	-53	-53	-61	-62	-48	-61
-59	-72	-58	-70	-77	-68	-73	-65	-59	-61
-56	-76	-68	-70	-75	-64	-70	-51	-72	-154
-123	-102	-86	-86	-91	-84	-88	-92	-94	-112
-121	-103	-39	73	70	-4	-69	-102	-112	-114
-108	-120	-116	-106	-114	-113	-131	-123	-129	-137
-121	-135	-135	-136	-158	-154	-153	-154	-149	-153
-142	-156	-144	-154	-153	-150	-134	-147	-226	-200
-160	-152	-157	-166	-167	-162	-168	-174	-158	-180
-182	-178	-186	-178	-197	-176	-179	-185	-174	-182
-169	-178	-176	-172	-182	-178	-182	-186	-177	-190
-194	-200	-204	-202	-212	-208	-199	-201	-199	-197
-192	-194	-203	-211	-225	-172	-152	-93	-55	-114
-153	-180	-173	-176	-183	-172	-184	-183	-172	-184
-194	-187	-188	-179	-187	-178	-188	-184	-187	-182
-178	-182	-194	-181	-182	-175	-177	-170	-166	-184
-182	-163	-171	-166	-163	-150	-164	-174	-150	-158
-163	-160	-153	-144	-140	-208	-222	-190	-158	-166
-158	-154	-167	-148	-158	-155	-150	-146	-147	-162
-155	-153	-161	-154	-170	-160	-157	-157	-136	-142
-144	-153	-146	-136	-147	-145	-142	-138	-137	-147
-153	-170	-166	-112	-9	64	22	-68	-98	-124
-123	-122	-111	-109	-212	-182	-136	-123	-126	-132
-124	-134	-135	-124	-130	-142	-138	-148	-149	-144
-145	-134	-147	-141	-147	-145	-145	-147	-137	-144
-144	-138	-141	-127	-149	-149	-134	-144	-144	-140
-138	-132	-150	-142	-134	-140	-134	-134	-122	-128
-124	-106	-138	-194	-175	-138	-115	-115	-130	-129
-122	-124	-137	-122	-137	-133	-131	-129	-130	-130
-134	-125	-136	-134	-128	-135	-125	-138	-127	-125
-156	-161	-159	-103	8	34	-6	-94	-125	-137
-135	-129	-132	-126	-120	-120	-116	-122	-108	-105
-96	-126	-170	-126	-98	-89	-93	-102	-106	-98
-105	-99	-118	-118	-120	-134	-133	-142	-142	-142
-142	-130	-142	-149	-147	-150	-151	-150	-150	-153
-163	-146	-148	-152	-147	-157	-150	-160	-160	-147
-148	-141	-149	-138	-142	-143	-135	-147	-124	-130
-225	-208	-166	-144	-151	-152	-155	-154	-142	-150
-144	-147	-150	-152	-150	-156	-155	-160	-160	-161
-182	-187	-178	-142	-53	-5	-23	-100	-140	-155
-169	-154	-156	-150	-141	-152	-138	-146	-131	-128
-149	-138	-147	-130	-134	-135	-140	-110	-139	-207
-174	-127	-126	-119	-123	-129	-126	-135	-123	-132
-137	-130	-132	-122	-113	-129	-118	-120	-110	-107
-104	-96	-100	-96	-90	-99	-88	-93	-97	-93
-108	-118	-125	-126	-129	-142	-132	-141	-141	-138
-139	-156	-161	-161	-165	-174	-157	-244	-281	-233
-218	-221	-236	-246	-244	-253	-273	-274	-279	-283
-314	-344	-342	-325	-270	-229	-229	-306	-352	-367

AUTHOR INDEX

SUBJECT INDEX

Printed in the United Kingdom
by Lightning Source UK Ltd.
102718UKS00001B/4-27

9 780863 411939

Adaptive Prediction and Predictive Control

Control often follows predictions: predictive control has been highly successful in producing robust and practical solutions in many real-life, real-time applications. Adaptive prediction covers a variety of ways of adding 'intelligence' to predictive control techniques. Many different groups, with widely varying disciplinary backgrounds and approaches, are tackling the same problem from different angles; these groups are sometimes unaware of alternative approaches from other disciplines.

This book attempts to give a unified and comprehensive coverage of the principles and methods that these groups have developed. It avoids basing its descriptions on very complex mathematical formulations but still gives a rigorous exposure to the subject, and illustrates the theory with many practical examples. It is chiefly aimed at students, researchers and practitioners, but will also be accessible to the non-specialist.

Partha Pratim Kanjilal is an Associate Professor at the Indian Institute of Technology, Kharagpur. He is a graduate of the IIT and of the University of Sheffield. His research interests include modelling and prediction of complex processes, linear and nonlinear time-series analysis, signal processing and predictive control. He has a particular interest in the applications of orthogonal control. Dr Kanjilal has previously worked as a research fellow at Oxford University, at CSIR and at Tata Steel in India. He has published over 25 refereed papers in journals and international conferences.

ISBN 978-0-86341-193-9

9 780863 411939

The Institution of Engineering and Technology
www.theiet.org
0 86341 193 2
978-0-86341-193-9